

Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

Diseño, impresión 3D, control por Arduino e
implementación en Robotstudio de un
brazo robótico

Autor: Daniel Dorante Ciscare

Tutores: David Muñoz de la Peña Sequeda

Ignacio Alvarado Aldea

Dep. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2017



Trabajo Fin de Grado
Grado de Ingeniería en Tecnologías Industriales

Diseño, impresión 3D, control por Arduino e implementación en Robotstudio de un brazo robótico

Autor:

Daniel Dorante Ciscares

Tutores:

David Muñoz de la Peña Sequedo

Ignacio Alvarado Aldea

Profesor titular

Dep. Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Trabajo Fin de Grado: Diseño, impresión 3D, control por Arduino e implementación en Robotstudio de un brazo robótico

Autor: Daniel Dorante Ciscares

Tutores: David Muñoz de la Peña Sequeda,
Ignacio Alvarado Aldea

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

A mi familia

A mis amigos

A mis maestros y profesores

Agradecimientos

En primer lugar quiero agradecer a David Muñoz e Ignacio Alvarado, mis tutores, por su apoyo, su paciencia y su comprensión. Sin su supervisión y su guía este trabajo no habría sido posible.

A mis maestros y profesores, por su esfuerzo y dedicación al ejercer su profesión, por dejarse la piel en formarme y transmitirme sus conocimientos.

A mis amigos, por su apoyo incondicional.

A mi familia, por su amor, por haberme enseñado a no rendirme jamás, por haber hecho de mí la persona que hoy soy, por haber creído en mí siempre, incluso cuando yo no lo hacía.

Por todo esto y mucho más,

Mil gracias.

Daniel Dorante Ciscare

Sevilla, 2018

Resumen

La redacción de este trabajo se ha realizado con el objetivo de diseñar un brazo robótico mediante programas de diseño gráfico y su posterior impresión 3D. Una vez hecho esto, se procede a controlar el brazo mediante el microcontrolador Arduino. Finalmente, se importa el robot y se implementa en el programa Robotstudio de la casa ABB.

Abstract

The writing of this work has been made with the aim of designing a robotic arm through graphic design programs and its subsequent 3D printing. Once this is done, the arm is controlled by the Arduino microcontroller. Finally, the robot is imported and implemented in the Robotstudio program of the ABB home.

Agradecimientos	9x
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Figuras	17
Índice de Planos	
1 Introducción	1
1.1. Motivación	1
<i>1.2. Impresión 3D. Aplicaciones</i>	<i>1</i>
<i>1.3. Arduino</i>	<i>2</i>
2 Diseño gráfico	5
<i>2.1. Introducción a Solid Edge</i>	<i>5</i>
2.1.1. Creación de una nueva pieza	5
2.1.2. Creación de un conjunto	6
2.1.3. Creación de un plano	7
<i>2.2. Articulaciones</i>	<i>10</i>
2.2.1. Base	11
2.2.2. Primera articulación	12
2.2.3. Segunda articulación	14
2.2.4. Garra	15
3 Control mediante Arduino	21
<i>3.1. Componentes</i>	<i>21</i>
3.1.1. Arduino UNO	21
3.1.2. Placa de pruebas	22
3.1.3. Servos	23
3.1.4. Joystick	25
<i>3.2. Programación</i>	<i>27</i>
3.2.1. Lenguaje y estructura	27
3.2.2. Programas	28
3.2.2.1. Control del brazo mediante Joysticks	28
3.2.2.2. Control del brazo mediante entrada de teclado	30
4 Robotstudio	33
<i>4.1. Introducción a Robotstudio</i>	<i>33</i>
4.1.1. Creación de una nueva estación	33
4.1.2. Importación de geometrías	34
4.1.3. Articulaciones y diseño de mecanismos	36
4.1.3.1. Base	36
4.1.3.2. Primera articulación	42
4.1.3.3. Segunda articulación	44
4.1.3.4. Garra	46

4.2. <i>Smart Components</i>	51
4.2.1. Señales, enlazamientos y diseños	55
4.3. <i>Controlador del robot</i>	57
5 Conclusiones	59
Referencias	60

ÍNDICE DE FIGURAS

Figura 1-1. Impresora DaVinci 2.0	2
Figura 1-2. Logo de Arduino	3
Figura 2-1. Inicio de Solid Edge	5
Figura 2-2. Plantilla de Solid Edge para crear una pieza	6
Figura 2-3. Plantilla de Solid Edge para crear un conjunto	6
Figura 2-4. Plantilla de Solid Edge para crear un plano	7
Figura 2-5. Ventana para cargar archivo	7
Figura 2-6. Ventana de opciones	8
Figura 2-7. Ventana de selección de orientación	8
Figura 2-8. Ventana para customizar la orientación	9
Figura 2-9. Ventana de selección de vistas	9
Figura 2-10. Ubicación de las vistas	10
Figura 2-11. Ejemplo de vistas en el plano	10
Figura 2-12. Pieza Circ-base	11
Figura 2-13. Pieza Soporte	11
Figura 2-14. Pieza Círculo	12
Figura 2-15. Base	12
Figura 2-16. Pieza Sujeta	13
Figura 2-17. Pieza Arti	13
Figura 2-18. Primera articulación	14
Figura 2-19. Pieza Arti-dos	14
Figura 2-20. Pieza Panel	15
Figura 2-21. Segunda articulación	15
Figura 2-22. Pieza Gripperservoplate	16
Figura 2-23. Pieza Sop-ser	16
Figura 2-24. Pieza Servoconnector	17
Figura 2-25. Pieza Gearsright	17
Figura 2-26. Pieza Gearsleft	18
Figura 2-27. Pieza Grippart	18
Figura 2-28. Pieza Liftparallelbar	19
Figura 2-29. Pieza Parallelbar	19
Figura 2-30. Garra	19
Figura 2-31. Brazo	20
Figura 3-1. Arduino UNO	22
Figura 3-2. Placa de pruebas	22

Figura 3-3. Placa de pruebas (zonas de alimentación)	23
Figura 3-4. Placa de pruebas (zona de conexiones)	23
Figura 3-5. Servo SM-S2309S	24
Figura 3-6. Servo Futaba S3003	24
Figura 3-7. Servo Sunfounder 55g	25
Figura 3-8. Servo Futaba S100	25
Figura 3-9. Joystick	26
Figura 3-10. Esquema de conexión del Joystick	26
Figura 3-11. Conexión Joystick-Arduino	27
Figura 3-12. Entorno de Arduino	27
Figura 3-12. Pestaña Herramientas	28
Figura 4-1. Inicio de Robotstudio	33
Figura 4-2. Estación vacía	34
Figura 4-3. Buscar geometría	34
Figura 4-4. Menú de búsqueda de archivos CAD	35
Figura 4-5. Geometría cargada en la estación	35
Figura 4-6. Brazo completo	35
Figura 4-7. Base	36
Figura 4-8. Crear mecanismo	36
Figura 4-9. Menú de creación de mecanismo	37
Figura 4-10. Ventana de creación de eslabón	37
Figura 4-11. Creación de mecanismos: eslabones creados	38
Figura 4-12. Ventana de creación de ejes	38
Figura 4-13. Creación de mecanismos: eje creado	39
Figura 4-14. Ventana de creación de sistema de coordenadas	39
Figura 4-15. Creación de mecanismos: sistema de coordenadas creado	40
Figura 4-16. Ventana de creación de calibración	40
Figura 4-17. Creación de mecanismos: calibración creada	41
Figura 4-18. Selección de movimiento de ejes de mecanismo	41
Figura 4-19. Movimiento de ejes de mecanismo	42
Figura 4-20. Conexión de piezas	42
Figura 4-21. Primera articulación	43
Figura 4-22. Primera articulación: eslabones creados	43
Figura 4-23. Primera articulación: eje de giro	43
Figura 4-24. Articulación girada 45°	44
Figura 4-25. Segunda articulación	44
Figura 4-26. Segunda articulación: eslabones creados	45

Figura 4-27. Segunda articulación: eje de giro	45
Figura 4-28. Articulación girada 45 °	45
Figura 4-29. Garra	46
Figura 4-30. Giro_garra: eslabones creados	46
Figura 4-31. Giro_garra: eje creado	47
Figura 4-32. Garra girada 45°	47
Figura 4-33. Cierra_garra: eslabones creados	47
Figura 4-34. Cierra_garra: primer eje (J1) creado	48
Figura 4-35. Cierra_garra: segundo eje (J2) creado	48
Figura 4-36. Ventana de creación de dependencias	49
Figura 4-37. Creación de dependencia	50
Figura 4-38. Garra cerrada	51
Figura 4-39. Creación de Smart Components	51
Figura 4-40. Ventana de creación de Smart Components	52
Figura 4-41. Definición de Cierra_Pinza	53
Figura 4-42. Definición de Abre_Pinza	53
Figura 4-43. Definición de Sensor_pinza	54
Figura 4-44. Ubicación de Sensor_pinza	54
Figura 4-45. Definición de Attacher y Detacher	55
Figura 4-46. Ventana para añadir señales de entrada/salida	55
Figura 4-47. Diseño de SC_Pinza	56
Figura 4-48. Propiedades de SC_Pinza: Cierra desactivada	56
Figura 4-49. Propiedades de SC_Pinza: Cierra activada	57
Figura 4-50. Creación de controlador	57

1 INTRODUCCIÓN

*La única manera de encontrar los límites de lo posible
es yendo más allá de ellos, en lo imposible*

- Arthur C. Clarke -

Este trabajo se redacta con el objeto de diseñar un brazo robótico mediante el programa de diseño SolidEdge, su impresión 3D y su control mediante el microcontrolador Arduino. Una vez hecho todo ésto se procede a implementar el brazo diseñado en el programa Robotstudio. Se realizará una presentación de SolidEdge y una guía de utilización de Robotstudio.

1.1. Motivación

La motivación de este trabajo de fin de grado surge de la voluntad de trabajar y conocer la impresión 3D, para mostrar sus usos y aplicaciones. En este caso se diseñará un pequeño brazo robótico, a semejanza de los robots industriales.

La realización de este trabajo ilustrará el uso de una herramienta de diseño gráfico, SolidEdge, con la que podremos diseñar las piezas que necesitemos para construir nuestro robot, así como la utilización del microcontrolador Arduino para poder controlar y manejar el brazo una vez impreso y construido.

Además, se implementará el brazo en el software Robotstudio, con lo que se aprovechará para hacer una guía del diseño de mecanismos y el uso de Componentes Inteligentes.

1.2. Impresión 3D. Aplicaciones

La impresión 3D es un grupo de tecnologías de fabricación por adición donde un objeto tridimensional es creado mediante la superposición de capas sucesivas de material. A continuación se enumeran algunos métodos de impresión: extrusión, hilado, granulado, laminado, etc.

Este trabajo se ha realizado con una impresora por extrusión, por lo que se comentará en qué consiste esta tecnología: un motor se encarga de mover el filamento a través de un elemento de calentamiento que lo funde a una temperatura que normalmente oscila entre 170 y 240 grados (dependiendo del tipo de material utilizado). El filamento emerge fundido del extrusor y rápidamente se endurece a temperatura ambiente para fijarse con la capa inferior. El cabezal de impresión (o la plataforma de construcción) se mueve por el plano XY (horizontal) antes de moverse en el eje Z (vertical) una vez que cada capa es completada. Así, el objeto es impreso capa a capa desde la inferior a la superior.

En concreto, este trabajo se ha realizado con la impresora DaVinci 2.0, de XYZPrinting, la cual incluye dos extrusores.

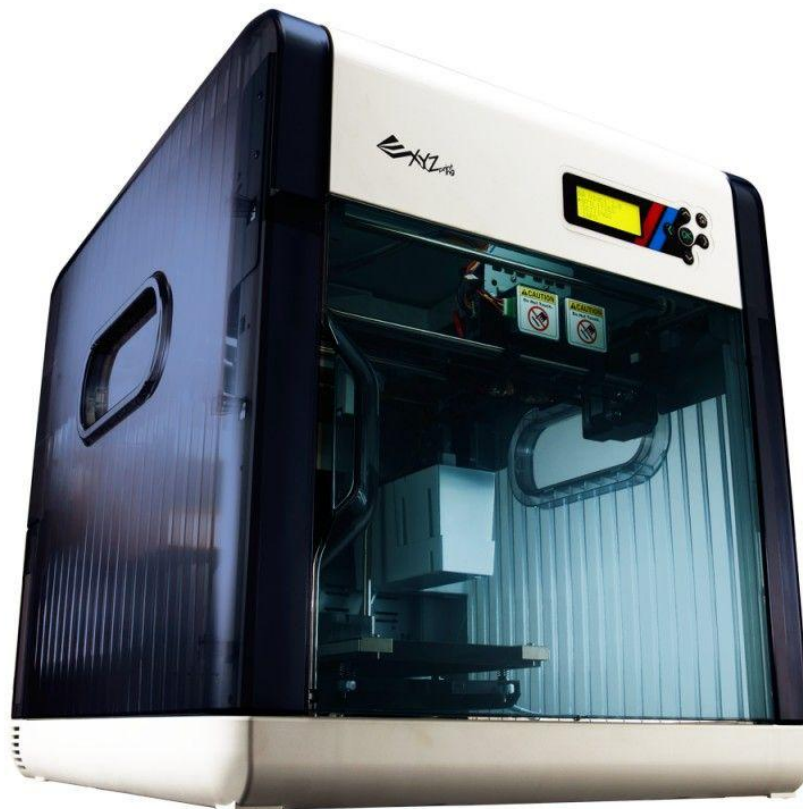


Figura 1-1. Impresora DaVinci 2.0

1.3. Arduino

Arduino es una compañía open source de hardware y software, así como un proyecto y comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan sensor y controlar objetos del mundo real.

Los diseños de las placas Arduino usan una variedad de microcontroladores y microprocesadores. Generalmente el hardware consiste de un microcontrolador sobre una placa de circuito impreso. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos

multidisciplinarios. Toda la plataforma, incluyendo sus componentes de hardware (esquemáticos) y Software, son liberados con licencia libre que permite libertad de acceso a ellos.

El proyecto Arduino comenzó en 2003 como un programa para estudiantes en el Interaction Design Institute Ivrea en Ivrea, Italia, con el objetivo de proporcionar una forma fácil y económica de que principiantes y profesionales crearan dispositivos que pudieran interactuar con su entorno mediante sensores y actuadores. La primera placa Arduino comercial fue introducida en el año 2005, ofreciendo un bajo costo económico y facilidad de uso para novatos y profesionales. Buscaba ser usada en el desarrollo de proyectos interactivos.



Figura 1-2. Logo de Arduino

2 DISEÑO GRÁFICO

*La mejor forma de predecir el futuro es crearlo.
Abraham Lincoln*

El diseño gráfico de nuestro brazo robótico se realizará mediante el programa Solid Edge, el cual es un programa parametrizado de diseño asistido por computadora de piezas tridimensionales. Permite el modelado de piezas de distintos materiales, doblado de chapas, ensamblaje de conjuntos, soldadura y funciones de dibujo en plano para ingenieros. Para realizar este trabajo se ha usado la versión 20.

2.1. Introducción a Solid Edge

2.1.1 Creación de una nueva pieza

Una vez instalado y ejecutado el programa, procedemos a elegir la opción ‘Solid Part’ dentro de la pestaña ‘Create’, tal y como se aprecia en la siguiente imagen:

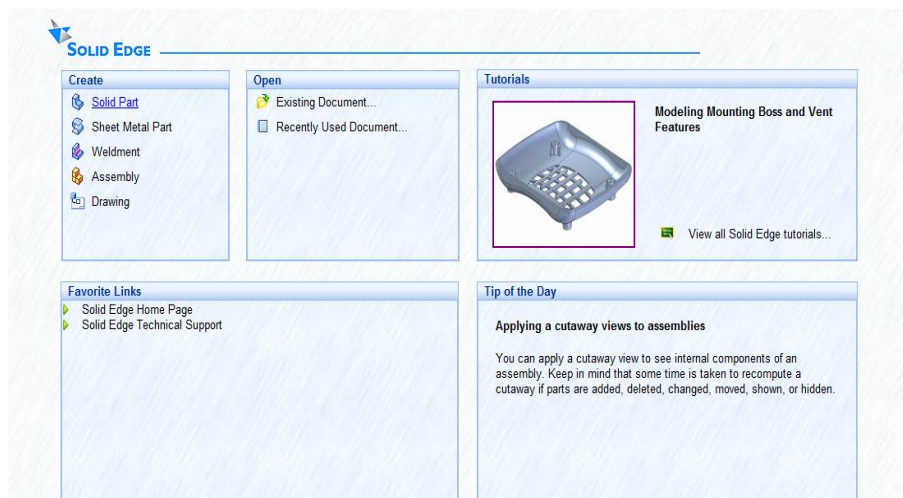


Figura 2-1. Inicio de Solid Edge

Una vez seleccionada la opción se nos mostrará la siguiente pantalla:

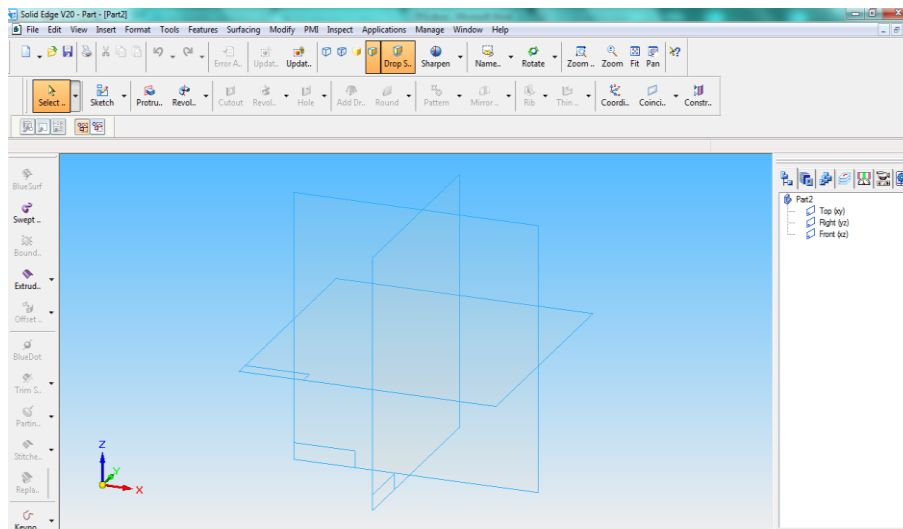


Figura 2-2. Plantilla de Solid Edge para crear una pieza

En esta ventana tenemos acceso a diversas herramientas y comandos con los que podremos diseñar la pieza que deseemos, tales como dibujar un sketch en un plano, protusiones, revoluciones, agujeros o extrusiones. Una vez diseñada la pieza podemos guardarla para usarla posteriormente cuando creemos las articulaciones

2.1.2 Creación de un conjunto

Una vez tengamos diseñadas las piezas que forman un conjunto, podemos diseñarlo. Si en la figura 2-1 elegimos la opción 'Assembly', perteneciente a la pestaña 'Create' podremos ver esta pantalla:

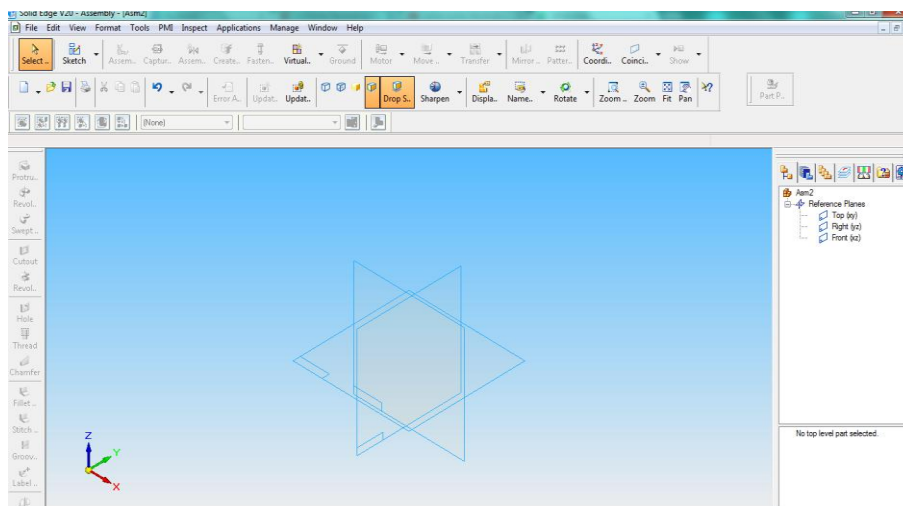


Figura 2-3. Plantilla de Solid Edge para crear un conjunto

Puede observarse que esta ventana es muy similar a la ventana de creación de pieza, pero ésta ofrece algunas nuevas funciones, como establecer las relaciones entre componentes. También puede verse que el menú que aparece a la derecha es distinto, y nos permitirá añadir las piezas que necesitemos para crear el conjunto, así como cualquier modificación de las relaciones existentes entre ellas.

2.1.3 Creación de un plano

Para cada pieza o conjunto que diseñemos, Solid Edge nos da la posibilidad de crear un plano con las vistas de la pieza que deseemos y en diferentes escalas. Si en la figura 2-1 seleccionamos la opción 'Drawing', perteneciente a la pestaña 'Create' se nos abrirá esta ventana:

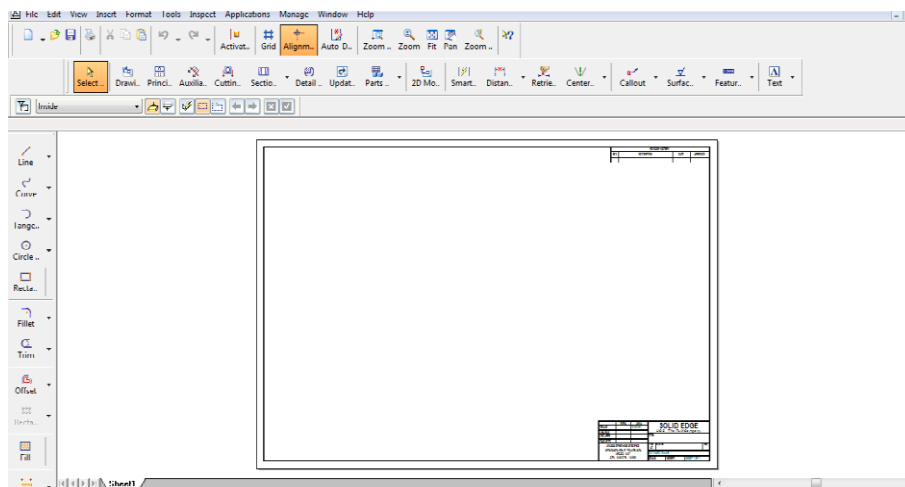


Figura 2-4. Plantilla de Solid Edge para crear un plano

El siguiente paso es cargar la pieza o el conjunto que deseemos, para lo cual debemos pulsar el botón 'Drawing View Wizard'. Una vez hecho esto nos aparecerá una ventana para cargar el archivo que queramos:

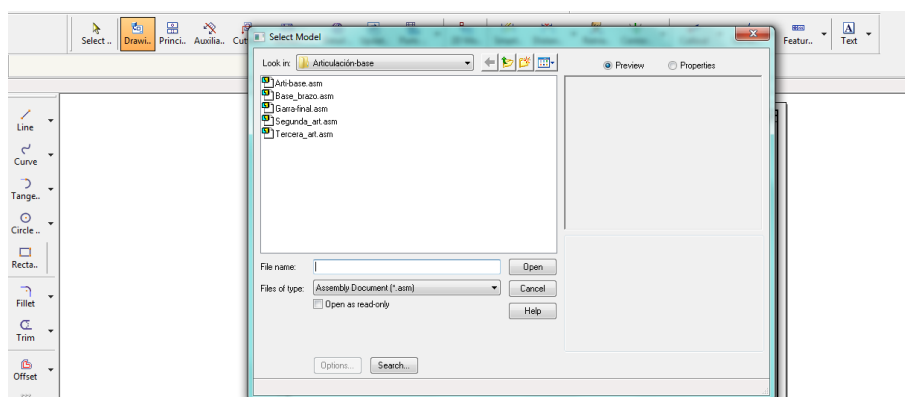


Figura 2-5. Ventana para cargar archivo

Una vez que seleccionemos un archivo, se nos abrirá una ventana de opciones:

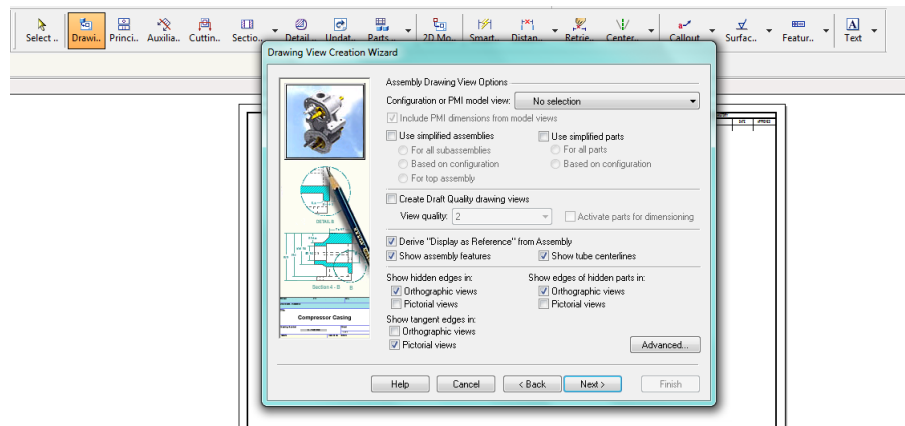


Figura 2-6. Ventana de opciones

En esta ventana no hemos cambiado nada de lo que viene predeterminado, así que pulsamos ‘Next’ y veremos lo siguiente:

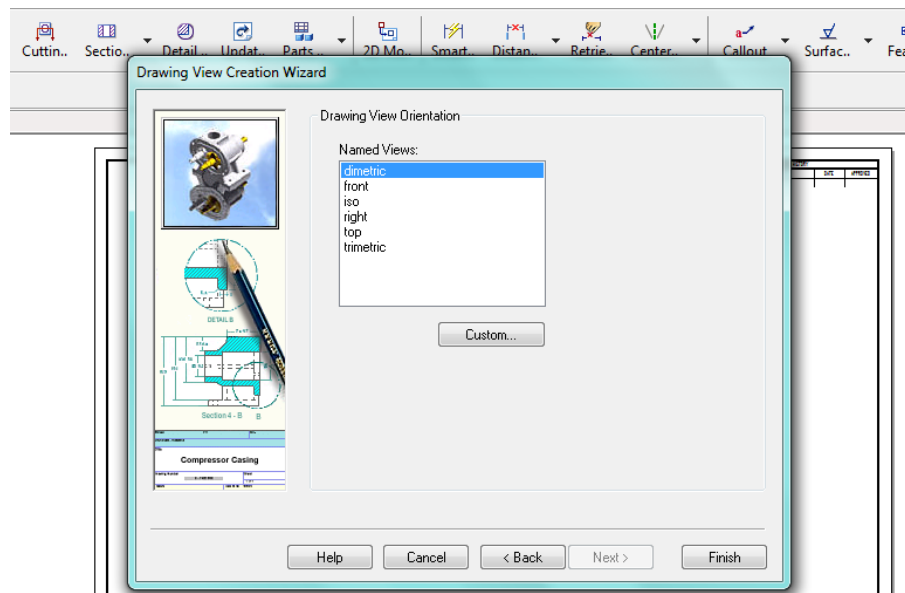


Figura 2-7 Ventana de selección de orientación

En esta ventana podemos elegir la orientación aplicada a la pieza, ya que en función de la que escojamos las vistas resultantes serán diferentes. Si queremos escoger una orientación a medida, podemos pulsar ‘Custom’ y se abrirá esta ventana:

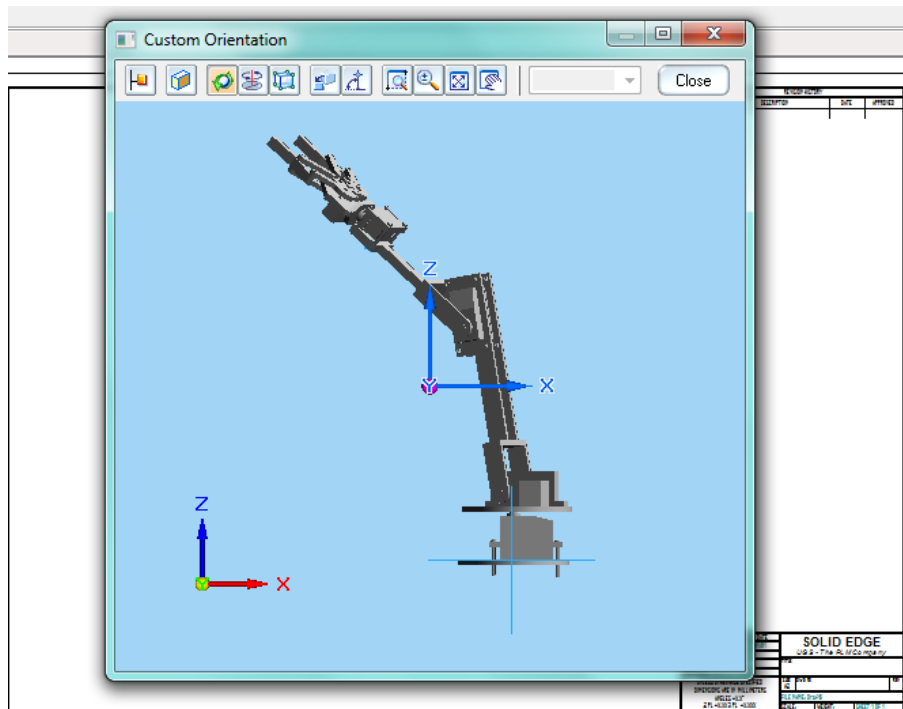


Figura 2-8 Ventana para customizar la orientación

En esta ventana podemos rotar la pieza, ya sea usando el ratón o introduciendo el ángulo numéricamente en el recuadro a la izquierda del botón 'Close'. Con ello conseguimos colocar la pieza en la orientación que queramos. Cuando hayamos elegido una orientación, pulsamos 'Close'. Ahora se nos mostrará lo siguiente:

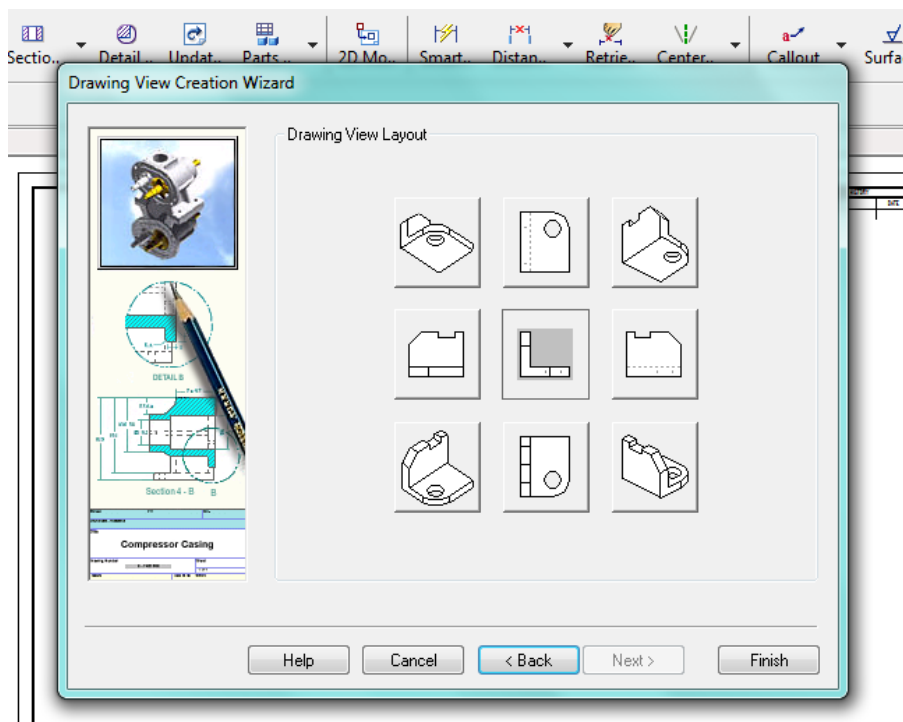


Figura 2-9 Ventana de selección de vistas

Aquí podemos seleccionar las vistas que queramos obtener de la pieza en cuestión, siendo el alzado la única vista obligatoria. Cuando hayamos elegido las vistas pulsamos 'Finish', lo que nos llevará de nuevo a la

plantilla de la imagen 2-4, donde podremos elegir donde queremos colocar las vistas:

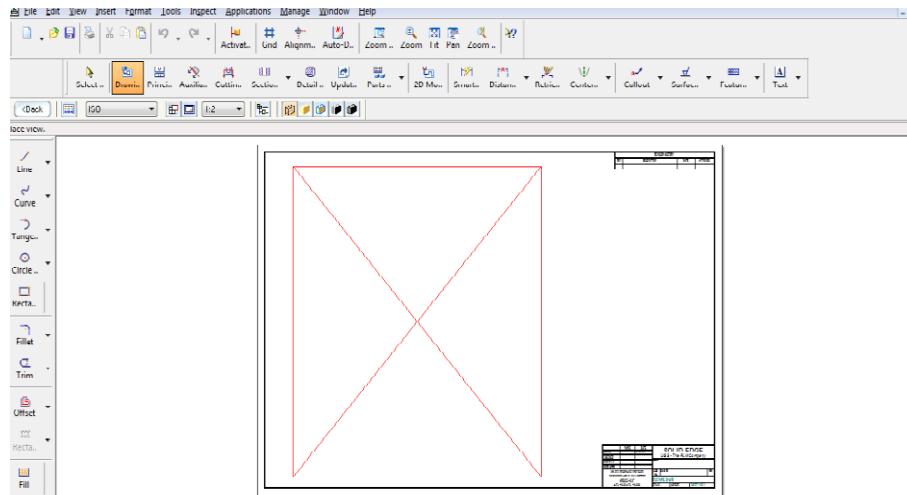


Figura 2-10 Ubicación de las vistas

Una vez que hayamos decidido dónde queremos colocar las vistas (que ocuparán el espacio indicado por el rectángulo rojo) haciendo click se colocarán las vistas, tal y como se presenta en esta imagen:

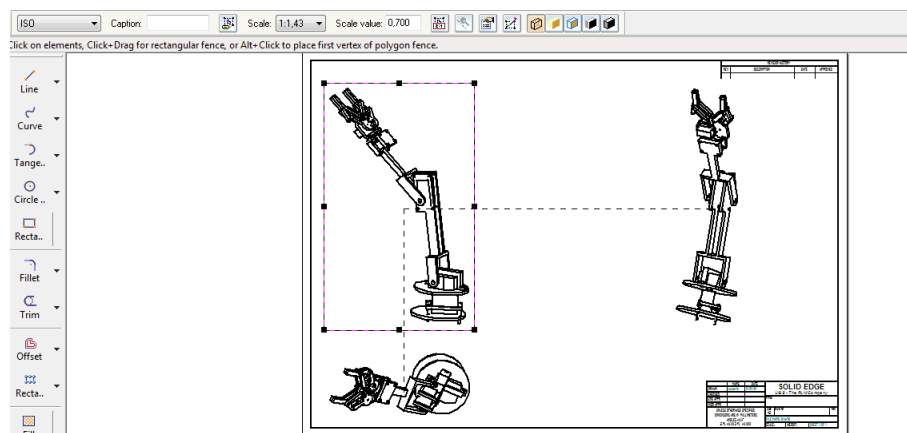


Figura 2-11 Ejemplo de vistas en el plano

Si hacemos click sobre una de las vistas podremos moverlas, y aparece una barra de opciones en la parte superior en la que podremos, entre otras cosas, cambiar la escala del plano.

2.2. Articulaciones

A continuación detallaremos las distintas articulaciones de nuestro brazo robótico, mostrando su diseño y las piezas de las que se componen.

2.1.1 Base

La base se compone de tres piezas, que se detallan a continuación:

-Circ-base: es un cilindro sobre el que se colocará el servo encargado de mover la base. Los cuatro orificios que presenta se usarán para poder atornillar esta pieza a la siguiente, el soporte del servo.

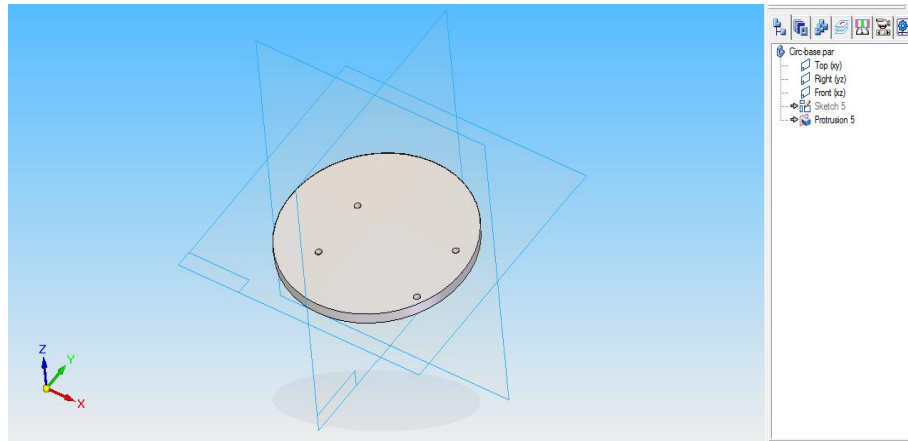


Figura 2-12 Pieza Circ-base

-Soporte: es la pieza que impedirá que el servo se mueva de su posición. La idea es la siguiente: se introduce al servo dentro del soporte, y éste se atornilla a la pieza Circ-base. Las medidas de esta pieza deben de ser muy precisas para poder ajustarse correctamente al servo: si el rectángulo interior es demasiado estrecho el servo no encajará, y si, por el contrario, es demasiado grande, el servo bailará y podrá moverse, lo que no es nada deseable.

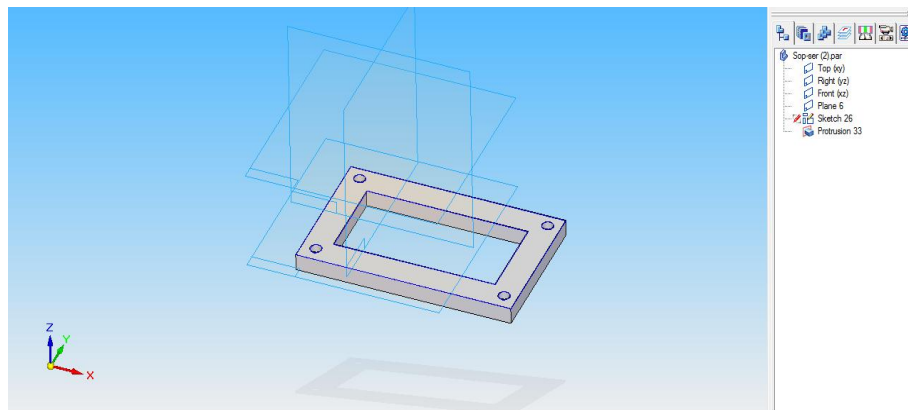


Figura 2-13. Pieza Soporte

-Círculo: es esta pieza la que se conectará el servo y, a su vez, sostendrá al resto del brazo. La circunferencia central servirá para introducir el extremo del servo, permitiendo así el movimiento de la pieza. El tamaño de esta circunferencia debe ser elegido con precisión, ya que, como nos ocurría antes, de no ser de tamaño adecuado no conseguiremos que el servo pueda moverla.

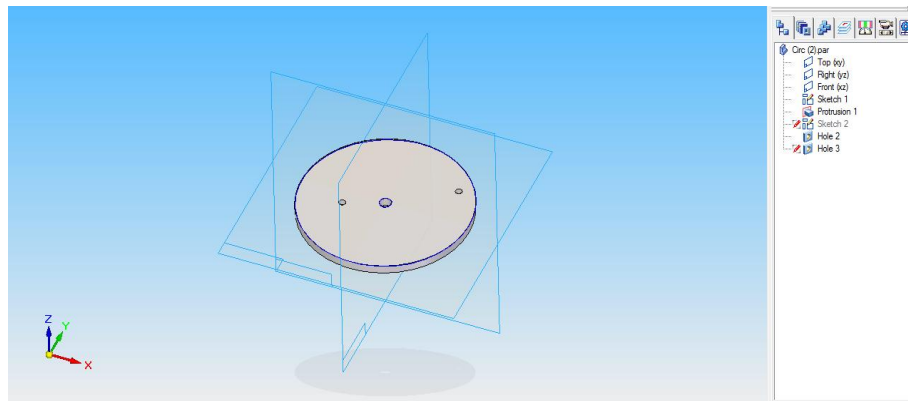


Figura 2-14. Pieza Círculo

La base queda:

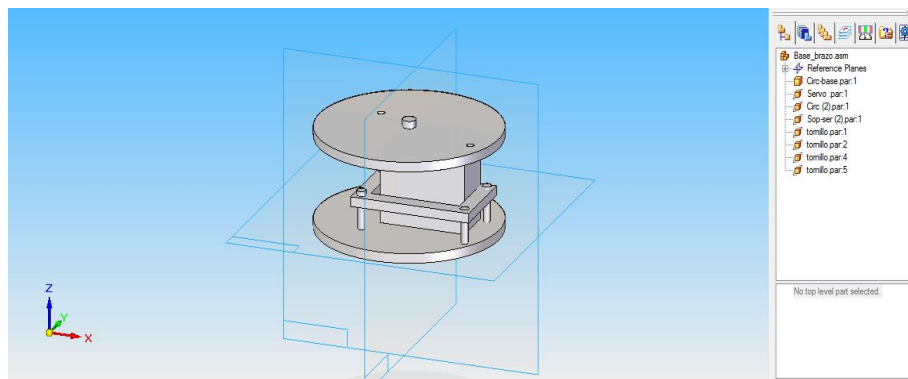


Figura 2-15. Base

Antes de pasar a la siguiente articulación comentaremos un problema que la base acabó presentando a posteriori. Manejando el brazo una vez impreso se pudo comprobar que pesaba demasiado, provocando que la base se levantara del suelo y el brazo se volcara. Para solucionar este problema pegamos la base a un tablón de madera, de manera que el brazo se mantenga siempre en su sitio gracias a que la base está fija permanentemente.

2.1.2 Primera articulación

A continuación se procederá de la misma manera que hicimos con la base: se detallarán cada una de las piezas que conforman la articulación, comentando su función y utilidad. Además se mostrarán el diseño y los planos de la pieza. Debemos mencionar que en esta articulación el servo encargado del movimiento de la misma se colocará tumbado sobre la pieza de la base “Círculo”, descrita anteriormente.

-Sujeta: esta pieza se colocará sobre el servo de la articulación, impidiendo que pueda moverse o deslizarse sobre la base, lo que no es permisible en absoluto. Los dos agujeros que presenta se usan para atornillar la pieza a la pieza “Círculo”.

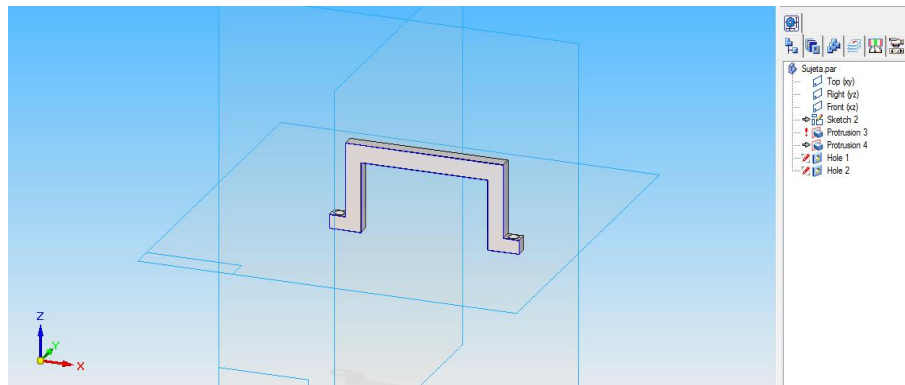


Figura 2-16. Pieza Sujeta

-Arti: esta pieza tiene dos funciones, siendo la primera de ellas la de conectarse al servo encargado de mover la articulación, que se colocará entre las dos patas inferiores. El extremo del servo se insertará en uno de los dos agujeros de las patas, permitiendo el movimiento de la pieza. La segunda función consiste en sostener al servo que moverá la siguiente articulación. Dicho servo se introducirá entre las dos aletas superiores, mediante el hueco rectangular que presentan.

Es necesario destacar una serie de cosas sobre esta pieza. La primera es que el hueco rectangular donde se introducirá el servo debe ajustarse con bastante precisión a éste, ya que de no hacerlo el servo no estará sujeto correctamente y podría deslizarse y/o caer. Lo segundo tiene que ver con un problema que hubo que afrontar a posteriori. Una vez impresa la pieza y montada, pude ver cómo el servo no conseguía mover la articulación, ya que pese a que estaba bien dimensionado, el extremo del servo giraba pero el movimiento no se transmitía a la pieza. La solución tomada finalmente fue la siguiente: conecté al extremo del servo un adaptador con dos aspas, el cual previamente había pegado a la pieza, de tal manera que la pieza está forzada a moverse al estar el adaptador pegado a ella. Esto trajo otro problema: al no contar con el espacio extra que ocupa el adaptador, el servo no cabía entre las dos patas de la pieza, por lo que se optó por cortar una de ellas, de manera que se consigue mover la pieza y el servo puede ser colocado donde le corresponde.

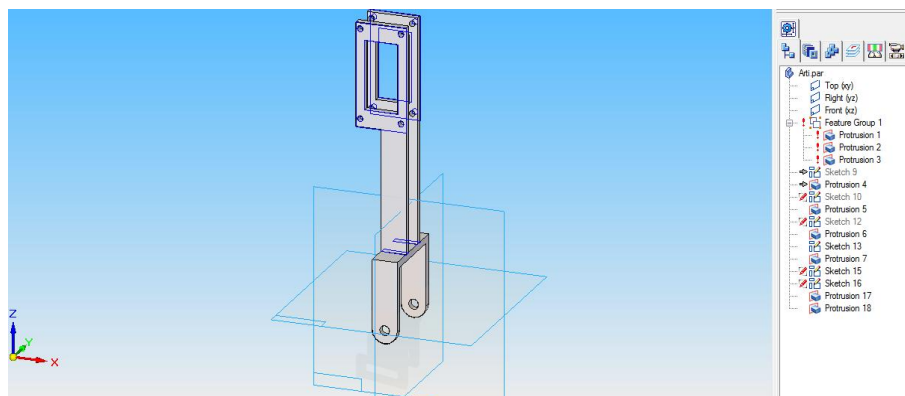


Figura 2-17 Pieza Arti

La articulación queda así finalmente:

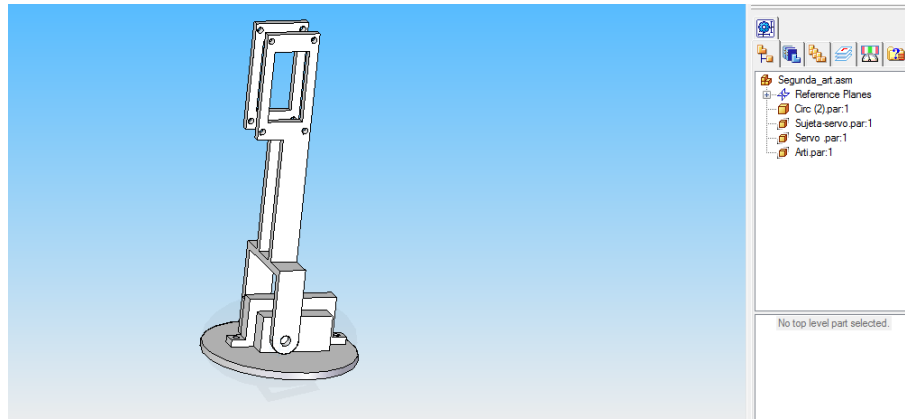


Figura 2-18 Primera articulación

2.1.3 Segunda articulación

Ahora vamos con la segunda articulación, que consta de dos piezas. Así pues vamos a mostrarlas y describirlas

-Arti-dos: en esta pieza se inserta el extremo del servo que sostenía la pieza anterior (Arti), en uno de los dos agujeros inferiores. Por otro lado, en el rectángulo superior se colocará otro servo (más pequeño que los usados anteriormente), que será el encargado de girar la garra.

Una vez impresa esta pieza nos pasó lo mismo que con la anterior (el servo no es capaz de moverla), y tomamos la misma solución: pegamos un adaptador a una de las patas para posteriormente colocárselo al servo y así poder mover la pieza, por lo que hubo que cortarle una de las patas para que pudiera introducirse el servo.

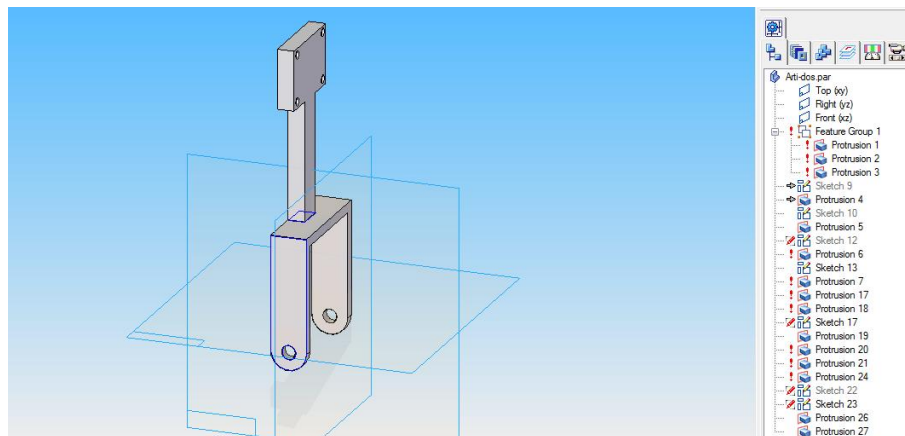


Figura 2-19 Pieza Arti-dos

-Panel: esta pieza se coloca sobre el servo encargado de rotar la garra, y paralelo al rectángulo superior de la pieza anterior. El objetivo es sujetar al servo e impedir su movimiento.

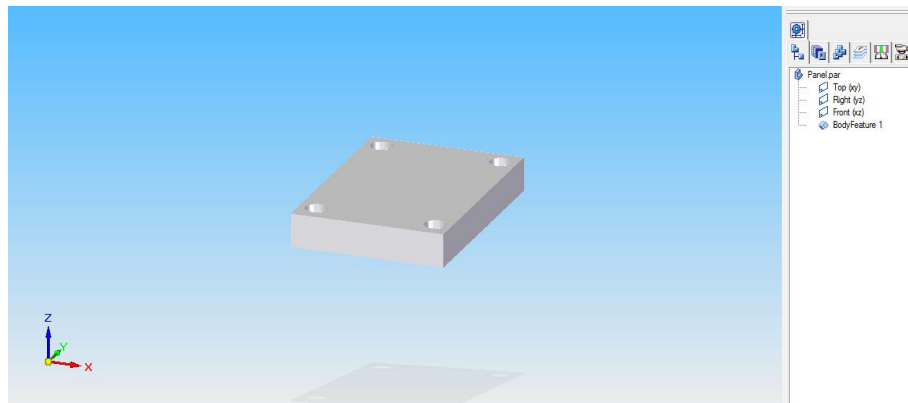


Figura 2-20 Pieza Panel

La articulación una vez montada queda:

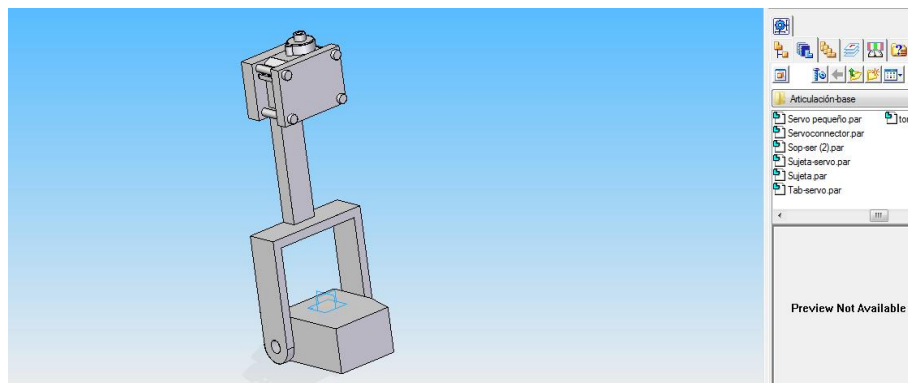


Figura 2-21 Segunda articulación

2.2.4 Garra

La garra es la parte del brazo más compleja ya que es la que más piezas distintas contiene. Vamos a ver todas y cada una de ellas para explicar y comprender su funcionamiento:

-Gripperservoplate: esta pieza es la que conforma el cuerpo de la garra. En el agujero rectangular se introducirá un servo, que será el encargado de abrir y cerrar la garra. Los dientes inferiores y los distintos agujeros que presenta se usarán para conectar el resto de piezas a ésta, y se comentarán cuando se estudie cada pieza en cuestión.

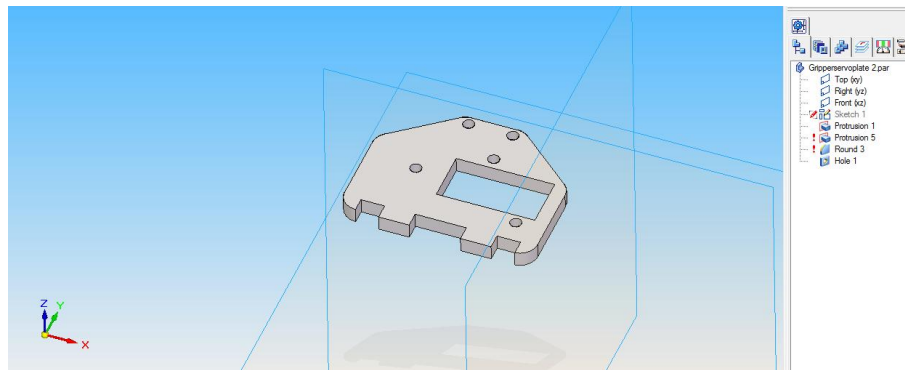


Figura 2-23 Pieza Gripperservoplate

-Sop-ser: esta pieza tiene como cometido sujetar al servo que ha de mover las piezas que permitan abrir y cerrar la garra. Debe colocarse debajo del agujero rectangular de la pieza anterior, conteniendo al servo en su interior y atornillándose a ella mediante los dos agujeros que presenta, haciéndolos coincidir con los dos agujeros circulares que se encuentran próximos a los lados mayores del rectángulo de Gripperservoplate. Debe diseñarse con precisión, de lo contrario no conseguirá encerrar al servo, lo cual tendría consecuencias inadmisibles.

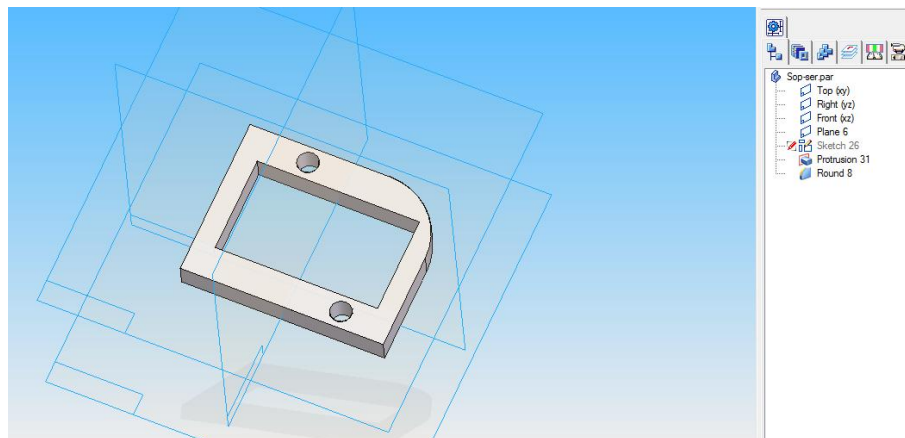


Figura 2-23 Pieza Sop-ser

-Servoconnector: esta pieza tiene una función doble, ya que se conecta al servo de la articulación anterior mediante el agujero circular que presenta en el centro, haciendo éste que gire con él, y por otro se conecta a la pieza anterior. La pieza presenta unos agujeros rectangulares, donde se introducirán los dientes rectangulares de la pieza anterior, de manera que al mover el servo la pieza ésta pueda transmitir el giro a la garra. Aquí podemos encontrarnos los problemas que hemos tratado anteriormente, es decir, si los agujeros no están diseñados con precisión no será posible que esta pieza cumpla con su función correctamente.

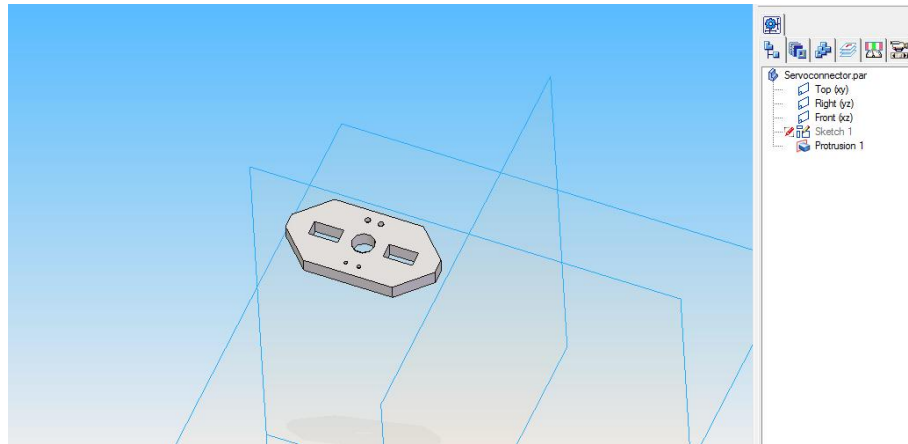


Figura 2-24 Pieza de Servoconnector

-Gearsright: esta pieza es de suma importancia, pues es la que se conecta al servo alojado en el hueco rectangular de la pieza Gripperservoplate , el cual hace girar la pieza y ésta transmite el movimiento a su compañera Gearsleft mediante los dientes que presenta. El extremo del servo se introduce en el agujero circular que tiene la pieza en su centro.

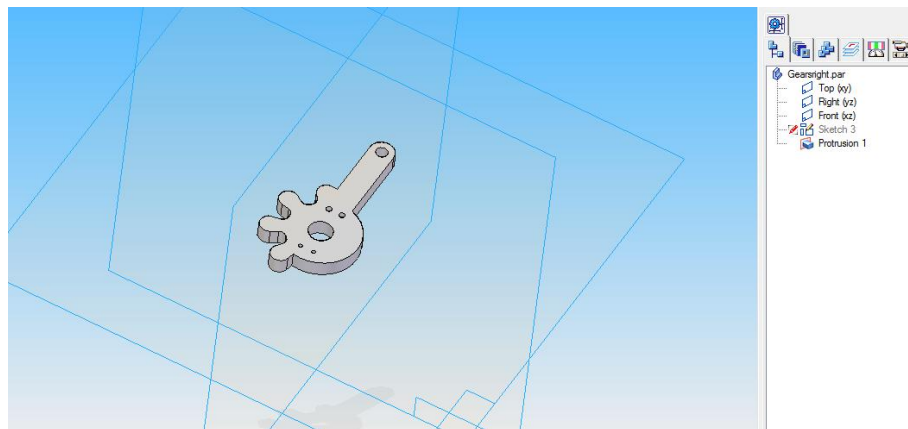


Figura 2-25 Pieza Gearsright

-Gearsleft: esta pieza es muy similar a la anterior, y debe encajarse con ella mediante el juego de dientes y huecos que ambas presentan. Además, debe atornillarse a la pieza Gripperservoplate, haciendo coincidir el agujero que ésta presenta a la izquierda del agujero rectangular para el servo con el agujero central que tiene la pieza que estamos tratando.

Cabe destacar que es de vital importancia que los dientes y los huecos de las dos últimas piezas deben encajar correctamente o no será posible abrir y cerrar la garra. Una vez impresas ambas, fue necesario limar un poco los dientes para garantizar que pudieran cumplir con su función.

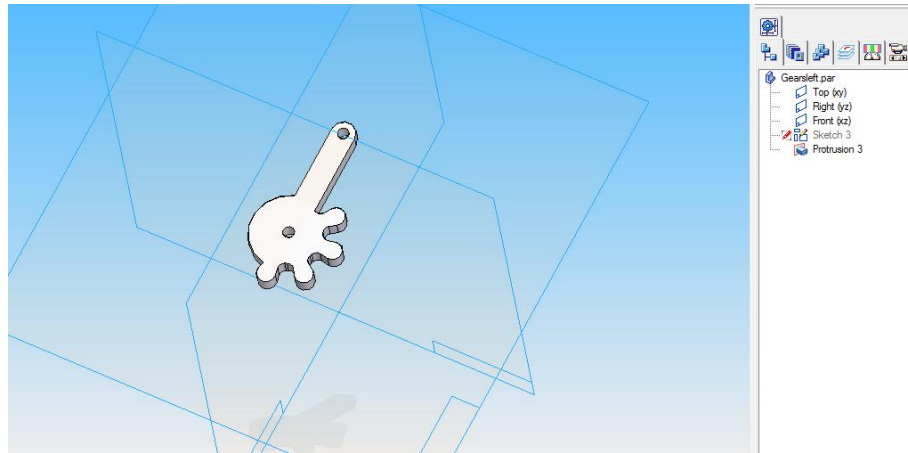


Figura 2-26 Pieza Gearsleft

-Grippart: necesitaremos cuatro piezas como ésta. Dos se atornillarán (mediante el agujero concéntrico con el borde redondeado) con Gearsleft y otras dos con Gearsright, usando el agujero que esas piezas presentan en su extremo. La función de estas piezas no es otra que la de efectuar el agarre sobre el objeto que se desee manipular, ya que al haber una pareja de piezas enfrentadas y conectadas a Gearsleft y Gearsright, que tienen giros de sentido contrario, se acercarán o alejarán en función de lo que deseemos, pudiendo coger o soltar un objeto.

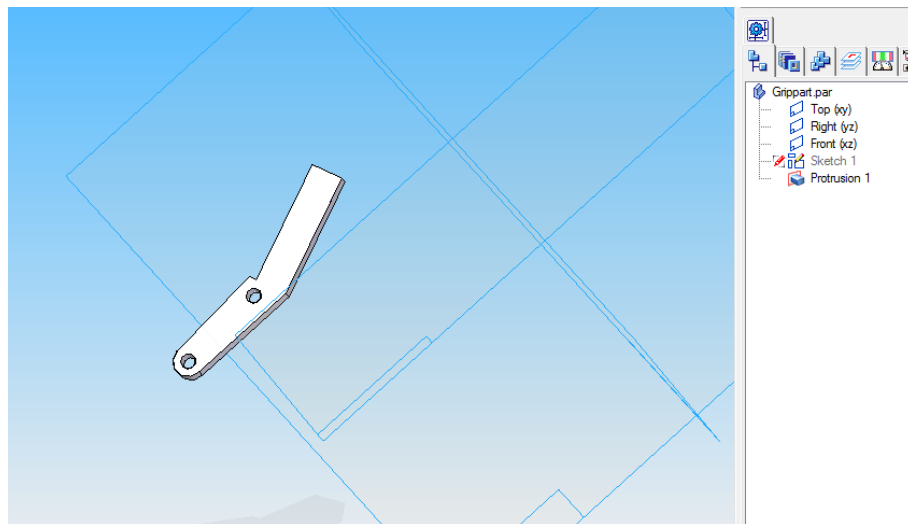


Figura 2-27 Pieza Grippart

-Liftparallelbar: esta pieza debe ser atornillada a la pieza Gripperservoplate, y se realiza mediante los dos agujeros circulares que ésta tiene en la parte superior. Esta pieza se coloca de manera intermedia entre Gripperservoplate y Parallelbar, que se estudiará a continuación.

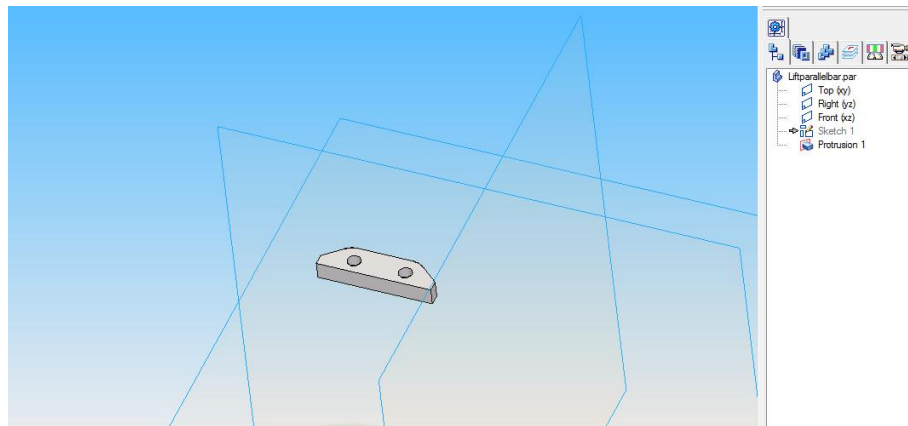


Figura 2-28 Pieza Liftparallelbar

-Parallelbar: será necesario imprimir dos piezas. Estas piezas tienen dos agujeros, y deben atornillarse a las piezas Grippart y Liftparallelbar. El objetivo de esta pieza es enlazar las dos piezas mencionadas anteriormente.

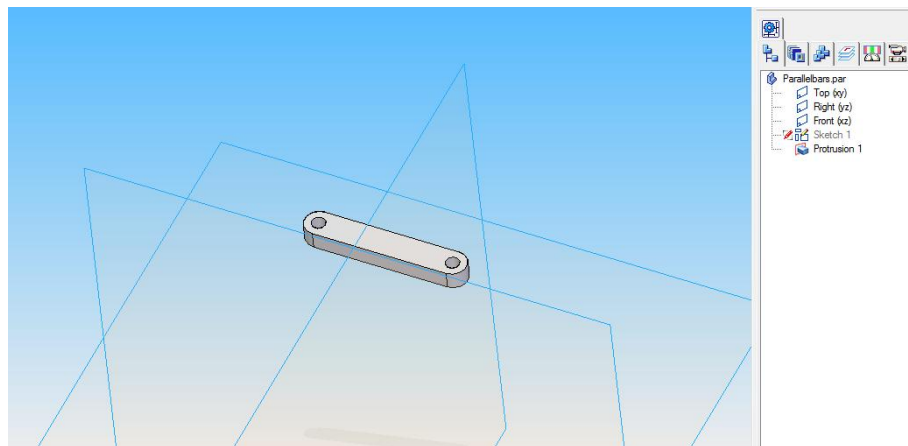


Figura 2-29 Pieza Parallelbar

Una vez diseñadas e impresas todas las piezas, la garra queda:

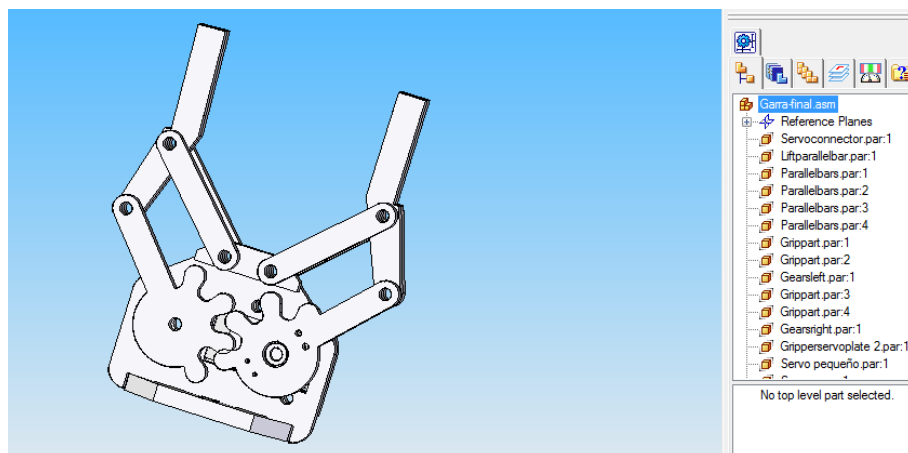


Figura 2-30 Garra

Aquí podemos ver el brazo una vez montado:

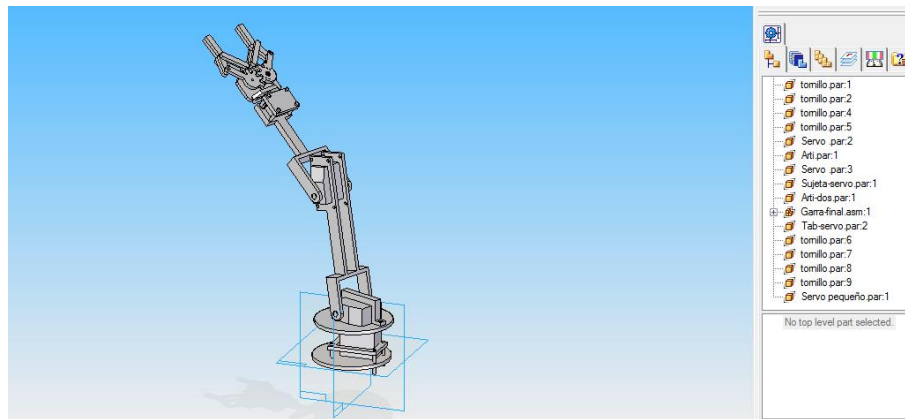


Figura 2-31 Brazo

3 CONTROL MEDIANTE ARDUINO

Vivir es enfrentar un problema tras otro. La forma en que lo encaras hace la diferencia

- Benjamin Franklin -

Para el control del brazo mediante Arduino se han realizado tres programas distintos, y cada uno de ellos utiliza distintas estrategias para cumplir con su cometido. Para abordar este capítulo se comenzará analizando los componentes utilizados en el proyecto, y posteriormente pasaremos a detallar cada uno de los programas utilizados, su funcionamiento, problemas que presentaron, etc.

3.1 Componentes

3.1.1 Arduino UNO

Este es el modelo de Arduino que usaremos, el cual consiste en una placa con un microcontrolador y con circuitería de soporte. Mediante el puerto USB podemos conectar la placa a un ordenador, desde el cual podremos programar el microcontrolador.

Dispone de 14 pines que pueden configurarse como entrada o salida y a los que puede conectarse cualquier dispositivo que sea capaz de transmitir o recibir señales digitales de 0 y 5 V. También dispone de entradas y salidas analógicas. Mediante las entradas analógicas podemos obtener datos de sensores, mientras que las salidas analógicas suelen utilizarse para enviar señales de control en forma de señales PWM.

Puede alimentarse directamente a través del propio cable USB o mediante una fuente de alimentación externa, como puede ser un pequeño transformador o, por ejemplo una pila de 9V.



Figura 3-1 Arduino UNO

3.1.2 Placa de pruebas

Una placa de pruebas (en inglés: protoboard o breadboard) es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares. Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí. Uno de sus usos principales es la creación y comprobación de prototipos de circuitos electrónicos antes de llegar a la impresión mecánica del circuito en sistemas de producción comercial.

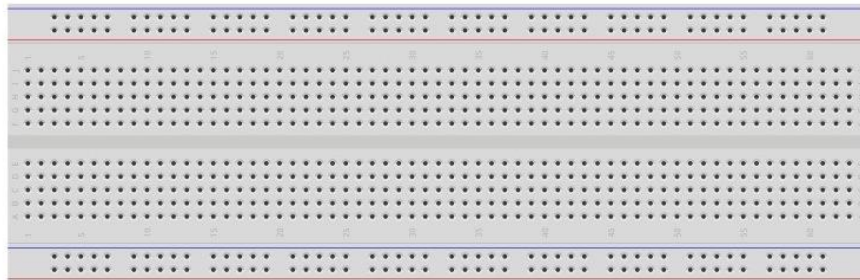


Figura 3-2 Placa de pruebas

La zona de alimentación está compuesta por orificios horizontales conectados entre sí eléctricamente a lo largo de toda la placa. Son dos líneas independientes; una para alimentación y otra para masa. Normalmente las placas de pruebas tienen dos zonas de alimentación situadas en lados opuestos para distribuir diferente alimentación. En la imagen, una de las zonas de alimentación se corresponde con las dos filas de orificios que se encuentran en la parte superior de la placa, mientras que la otra zona está en la parte inferior, con idéntica distribución (dos filas de orificios)

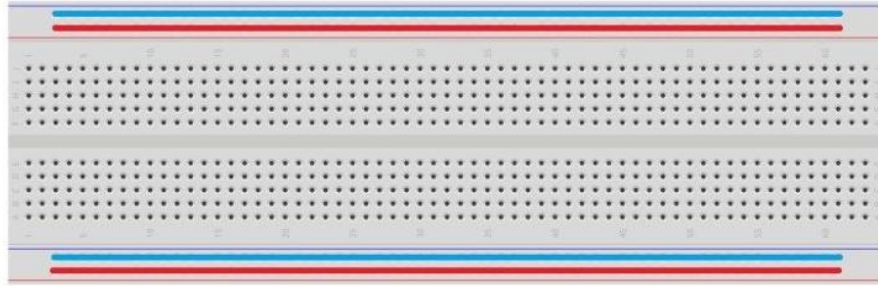


Figura 3-3 Placa de pruebas (zonas de alimentación)

La zona de conexiones está compuesta por columnas de orificios conectados eléctricamente entre sí. Cada columna es independiente eléctricamente con las demás, es decir, los orificios solo están conectados de forma vertical. Será en esta zona donde conectaremos los componentes electrónicos que usaremos para armar nuestro circuito. Cabe destacar que hay dos zonas de conexiones claramente diferenciadas, estando una de ellas en la parte superior y otra en la inferior. En la imagen puede verse, a modo de ejemplo, cómo están conectados eléctricamente los orificios correspondientes a tres columnas pertenecientes a la zona de conexiones superior.

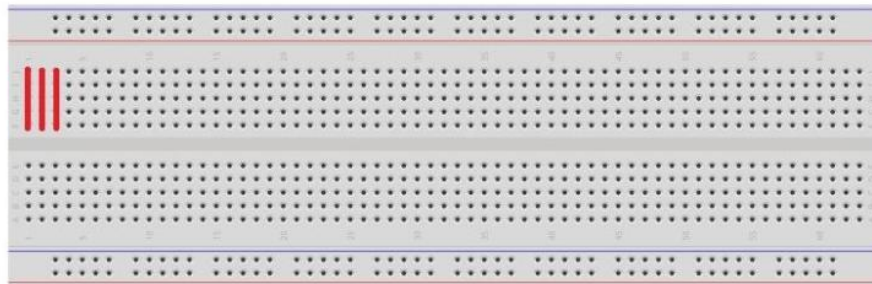


Figura 3-4 Placa de pruebas (zona de conexiones)

3.1.3 Servos

A continuación pasaremos a hablar de los distintos servos que se ocuparán de mover y controlar el brazo:

• *SM-S2309S*

Este servo es el de menor tamaño de todos los que usaremos. Vamos a necesitar dos de ellos, y se encargarán del cierre y el giro de la garra. Al ser los servos que menos fuerza tienen, se ocupan de las funciones que menos esfuerzo requieren, ya que están colocados al final del brazo y no deben mover más peso que el de la garra.



Figura 3-5. Servo SM-S2309S

· Futaba S3003

Este servo es de mayor tamaño que los dos anteriormente citados, ya que es necesario que tenga más fuerza. Lo usaremos para controlar el giro del brazo, por lo que pertenece a la base del brazo y debe soportar el peso de éste.



Figura 3-6. Servo Futaba S3003

· Sunfounder 55g

Este servo es aproximadamente del mismo tamaño que el mencionado anteriormente, y es uno de los que más peso debe soportar, pues es el encargado de mover la garra y las dos articulaciones (esto es, debe mover el brazo completo exceptuando la base, sobre la que se coloca).



Figura 3-7. Servo Sunfounder 55g

•Futaba S100

Este servo se coloca al final de la primera articulación y al comienzo de la segunda, haciendo de nexo entre ellas. Debe tener fuerza, ya que necesita mover una articulación y la garra

3.1.4 Joystick

Un joystick es un elemento de entrada para programas digitales. Internamente los joysticks están formados por un sistema de balancín con dos ejes ortogonales acoplados a dos potenciómetros. Estos potenciómetros realizan la medición de la posición de la palanca en ambos ejes. Por otro lado, uno de los ejes está apoyado en un microrruptor, lo que permite detectar la pulsación de la palanca.

Por tanto, los joysticks facilitan una señal analógica para la posición de cada eje, más una señal digital para la detección de la pulsación del mando. Cabe destacar que usaremos dos Joysticks para controlar todo el brazo.

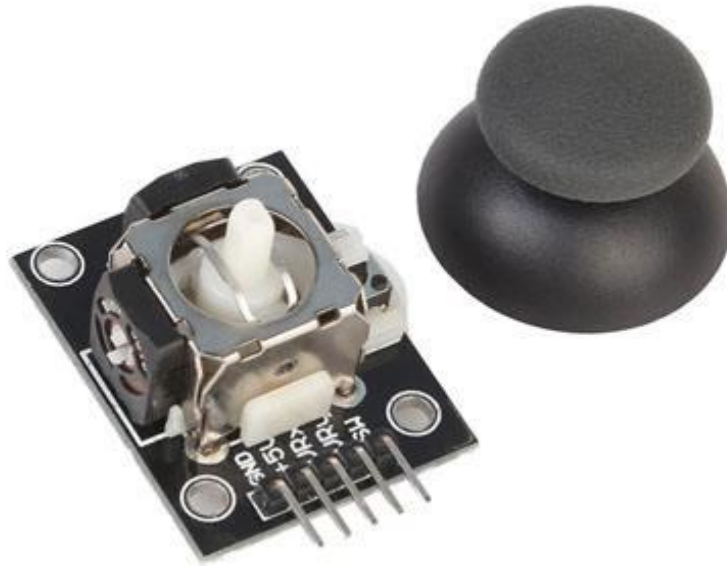


Figura 3-9 Joystick

El montaje de este tipo de dispositivos es sencillo. En primer lugar, alimentamos el módulo conectando Vcc y GND, respectivamente, a 5V y GND de Arduino. Para la medida analógica en ambos ejes, conectamos las salidas VRx y VRy a dos entradas analógicas de Arduino. Finalmente, si también queremos la señal digital del pulsador, conectamos la salida SW a una entrada digital de Arduino.

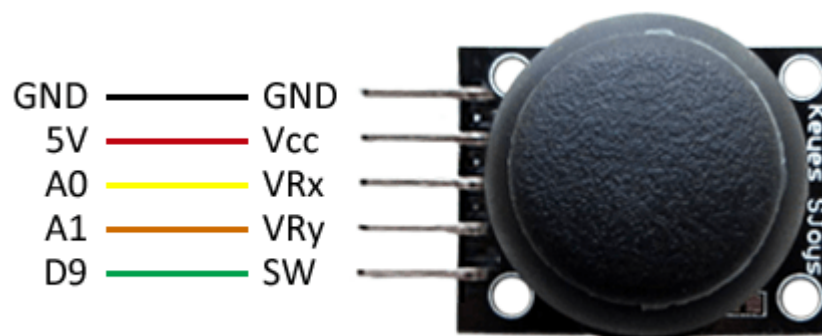


Figura 3-10 Esquema de conexión del Joystick

Mientras que el montaje, visto desde Arduino, sería el siguiente:

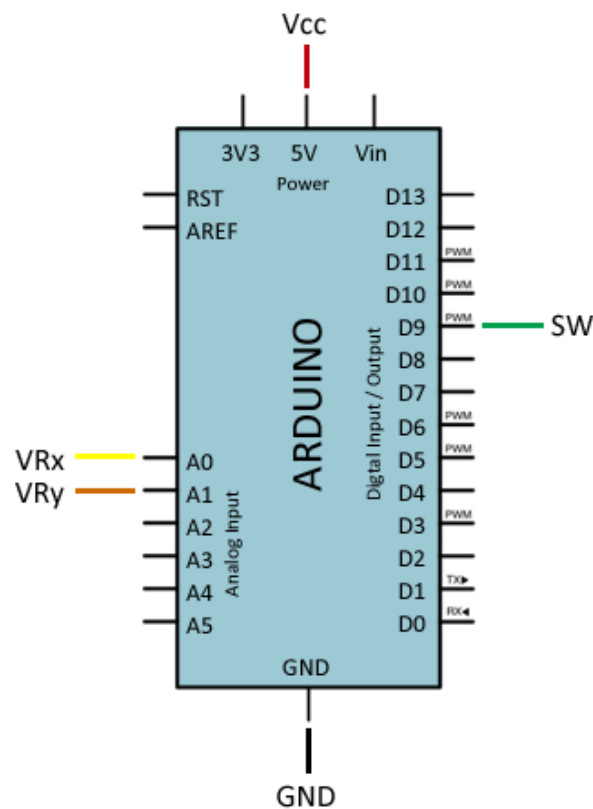


Figura 3-11 Conexión Joystick-Arduino

3.2 Programación

3.2.1 Lenguaje y estructura

Para programar en Arduino utilizaremos un lenguaje basado en C++. Una vez que iniciamos el programa nos encontramos en la siguiente ventana:



Figura 3-12 Entorno de Arduino

Una vez aquí debemos irnos a la pestaña “Herramientas” y seleccionar en la opción “Tarjeta” el modelo de

Arduino que vamos a usar:

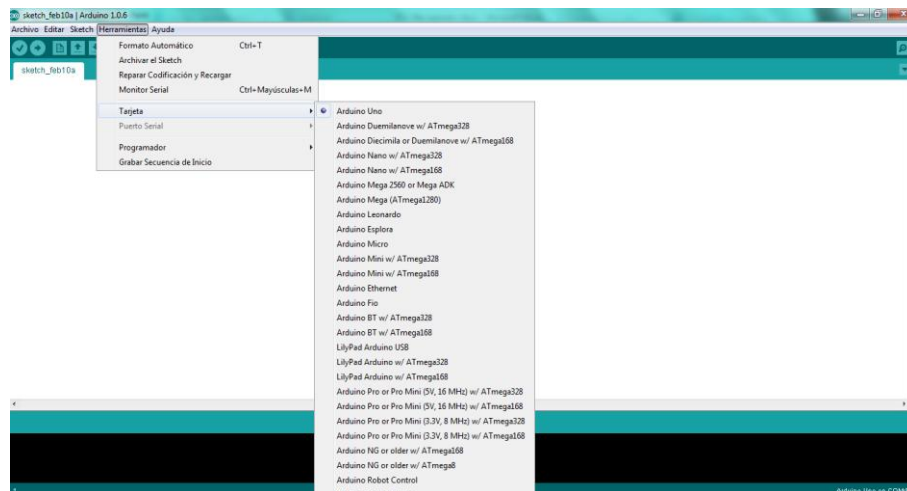


Figura 3-13 Pestaña Herramientas

En la opción inmediatamente inferior, “Puerto Serial” debemos elegir un puerto serie al que conectaremos nuestro Arduino.

A continuación pasaremos a hablar de la estructura de un programa en Arduino. La estructura básica del lenguaje de programación de Arduino se compone de al menos dos partes:

```
void setup()
{
  estamentos;
}
void loop()
{
  estamentos;
}
```

Setup() es la parte encargada de recoger la configuración y loop() es la que contienen el programa que se ejecutará cíclicamente. Debe contener la declaración de las variables, es la primera función a ejecutar en el programa, se ejecuta sólo una vez, y se utiliza para configurar el modo de trabajo de las E/S, configuración de la comunicación en serie, etc. Esta función se invoca una sola vez cuando el programa empieza, y debe ser incluido en un programa aunque no haya declaración que ejecutar.

La función bucle (loop) contiene el código que se ejecutara continuamente (lectura de entradas, activación de salidas, etc). Esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo. Después de llamar a setup(), la función loop se ejecuta de forma cíclica, lo que posibilita que el programa esté respondiendo continuamente ante los eventos que se produzcan en la tarjeta.

Ahora procederemos a explicar los tres programas realizados.

3.2.2 Programas

3.2.2.1 Control del brazo mediante Joysticks

En este programa usaremos los dos Joysticks para manejar el brazo. Con uno de ellos controlaremos la base y la primera articulación, y con el otro la segunda y tercera articulación. Además, ya que cada joystick dispone de un pulsador usaremos el de uno de ellos para abrir la garra y el otro para cerrarla.

Cabe destacar que en los tres programas necesitaremos incluir la librería “<Servo.h>”, que nos permitirá trabajar con los servos mediante una serie de comandos específicos. Vamos a explicar una parte del código que tiene especial relevancia:

```
int p1=analogRead(A0);
int p2 = analogRead(A1);
estado=digitalRead(12);

if ( p1 < 400 )           // Si la lectura es menor de 400
    angulo = angulo - salto ; // disminuimos el angulo
else if (p1>600)          // Si mayor de 600
    angulo = angulo + salto ; // Aumentamos el angulo
servo1.write(angulo);      // Y este es el que mueve el servo
delay (50);
// Este delay regula la velocidad del movimiento

if ( p2 < 400 )           // Si la lectura es menor de 400
    angulo2 = angulo2 + salto ; // disminuimos el angulo
else if (p2>600)          // Si mayor de 600
    angulo2 = angulo2 - salto ; // Aumentamos el angulo
servo2.write(angulo2);     // Y este es el que mueve el servo
```

Donde p1 y p2 son dos variables con los que leeremos las entradas analógicas de uno de los joysticks. Las entradas correspondientes al eje x son leídas por p1, mientras que p2 se encarga de las entradas correspondientes al eje y. Puede observarse que p1 está asociada al servo1 y p2 está asociada al servo2. Ya que las entradas del joystick en cada eje varían desde 0 a 1024 , se establece una zona muerta entre 400 y 600, en las que no se producirá movimiento alguno. Si se lee una entrada que sea menor que 400 o mayor que 600, al valor actual del ángulo del servo se le resta o se le suma una cantidad llamada salto, (en el programa definida con un valor de 3). Una vez actualizado el valor de ángulo, se mueve el servo a la posición definida con el nuevo ángulo, mediante la función “servo.write(angulo)”. Con esto se consigue lo siguiente: siempre que movamos el joystick a una posición que se encuentre fuera de la zona muerta, el ángulo se actualizará y el servo en cuestión se moverá en consecuencia, permitiéndonos controlarlo.

Vamos a hablar ahora de otra parte del código, correspondiente al cierre y apertura de la garra:

```
if (digitalRead(cierra)==0)
{
    angulo5=angulo5-salto2;
    servo5.write(angulo5);
}

if (digitalRead(abre)==0)
{
    angulo5=angulo5+salto2;
    servo5.write(angulo5);
}
```

```
}
```

Donde “abre” y “cierra” son dos variables asociadas al pulsador de cada joystick. Si pulsamos el joystick al que está asociado “cierra”, el ángulo del quinto servo (el servo que abre y cierra la garra) es disminuido una cantidad llamada salto2, con un valor de 10. A continuación se mueve el servo a la posición definida por el nuevo ángulo. Haciendo esto, la garra se cerrará progresivamente siempre y cuando tengamos pulsado el joystick encargado del cierre de la garra.

De similar manera funciona la apertura. Al pulsar el joystick al que está asociado “abre”, el ángulo del quinto servo es aumentado la cantidad salto2 y se mueve el servo a la posición definida por el ángulo actualizado. Por tanto, basta con dejar pulsado este joystick para que la garra se abra progresivamente.

3.2.2.2 Control del brazo mediante entrada de teclado

En este programa se definirán una serie de posiciones del brazo, y el programa nos permitirá cambiar de una a otra mediante una entrada de teclado. Las posiciones se definen asignando valores a los distintos ángulos de los servos, de la siguiente manera:

```
//Posición p
int angulo= 90 ;
int angulo2= 20;
int angulo3=0;
int angulo4=45;
int angulo5=45;
```

Como podemos ver, la posición p se ha definido dándole un valor a cada uno de los ángulos del brazo.

Vamos a ver ahora una parte del código en la que se puede apreciar el funcionamiento del programa:

```
//Si pulsamos p
if (entrada==112)
{
servo1.write(angulo);
delay(1000);
servo2.write(angulo2);
delay(1000);
servo3.write(angulo3);
delay(1000);
servo4.write(angulo4);
delay(1000);
servo5.write(angulo5);
delay(1000);
}
```

Donde “entrada” es una variable con la que se leerá la entrada por teclado. Si pulsamos la tecla “p”, se irá ordenando a cada servo que se mueva hacia la posición p definida anteriormente. Cabe destacar que las entradas se introducen mediante el monitor serie, el cual traduce automáticamente las entradas a código ASCII. Es por esto que la variable “entrada” es de tipo entero, ya que lee valores numéricos y no caracteres. Si “entrada” vale 112 (que se corresponde con p en código ASCII) quiere decir que se ha pulsado la tecla “p”, y por tanto se debe proceder al movimiento del brazo para alcanzar la posición requerida.

4 ROBOTSTUDIO

El poder de la imaginación nos hace infinitos

-John Muir-

El programa Robotstudio es un software de programación de robots industriales, diseñado y patentado por la empresa ABB, que permite un abanico de posibilidades en el mundo de la automatización industrial y la robótica manipuladora gracias a la versatilidad de su entorno de simulación y del lenguaje RAPID.

El objetivo que nos planteamos es el siguiente: una vez que hemos diseñado nuestro brazo y lo hemos controlado, trataremos de importarlo a Robotstudio para poder simular su funcionamiento en un espacio de trabajo. La versión utilizada de Robotstudio ha sido la 6.05.

4.1 Introducción a Robotstudio

4.1.1 Creación de una nueva estación

Una vez que ejecutamos el programa nos aparecerá la siguiente ventana de inicio, en la que debemos seleccionar “Crear una nueva estación vacía”

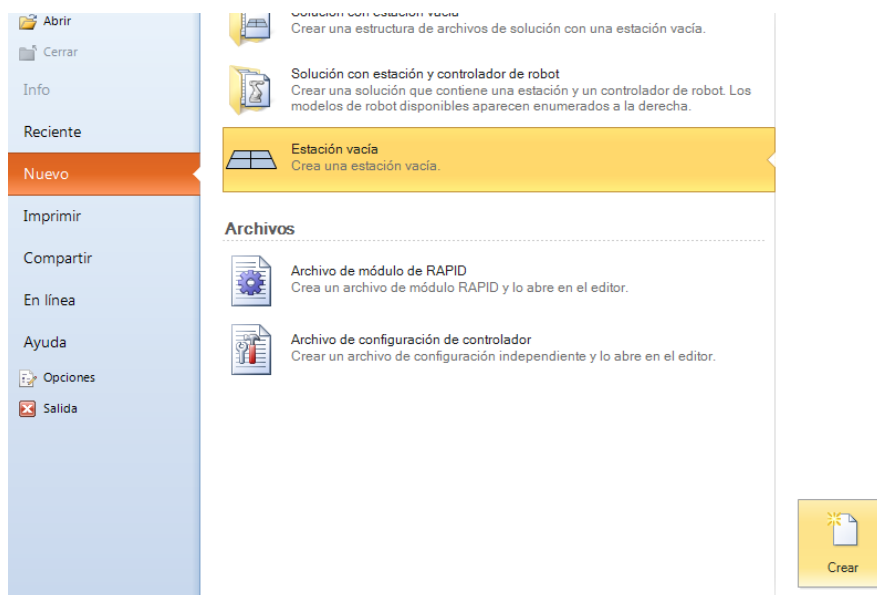


Figura 4-1. Inicio de Robotstudio

Una vez creada la estación nos encontraremos la siguiente ventana:

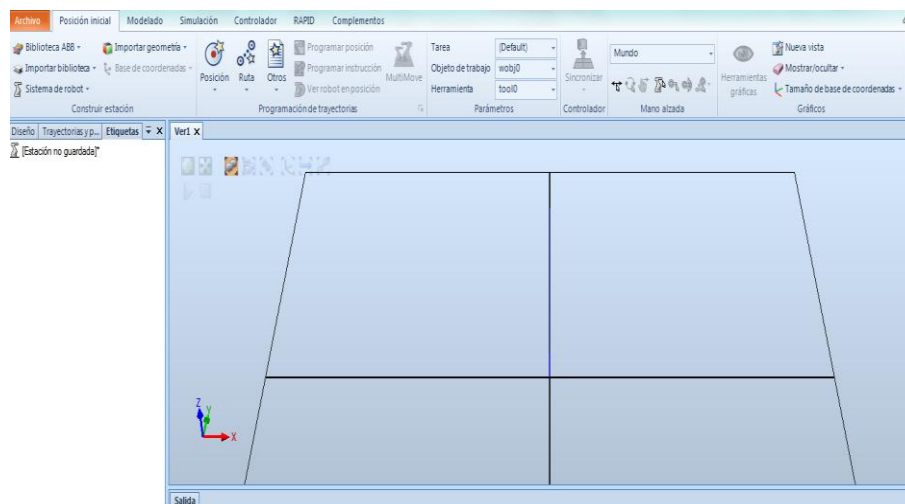


Figura 4-2. Estación vacía

4.1.2 Importación de geometrías

Ahora que tenemos una estación vacía creada, el siguiente paso será poder importar nuestro robot para poder trabajar con él. Para ello, debemos desplazarnos a la pestaña “Modelado” y desplegar las opciones de “Importar geometría”, tras lo cual debemos hacer click en “Buscar geometría”. Esta función nos permite buscar un archivo de CAD y añadirlo a nuestra estación.

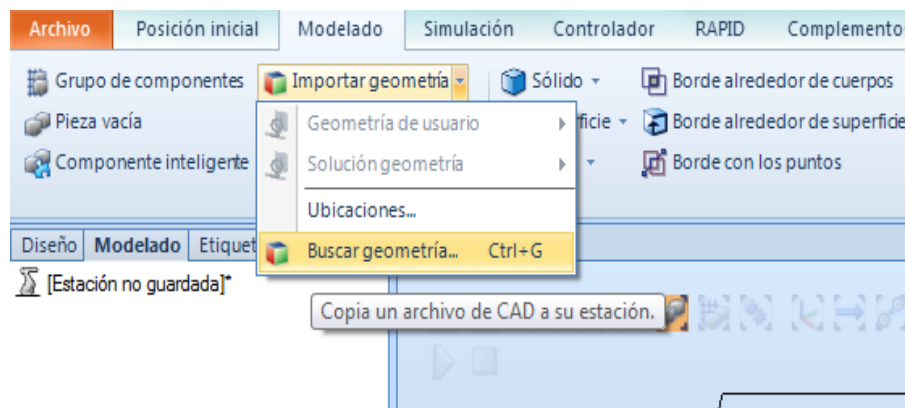


Figura 4-3. Buscar geometría

Una vez seleccionada esa opción se nos abrirá un menú en el que podremos buscar el archivo CAD que queramos subir:

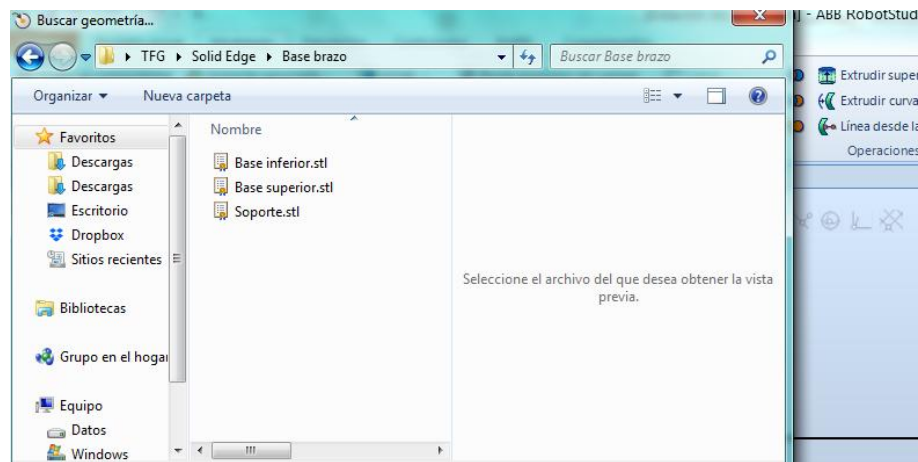


Figura 4-4. Menú de búsqueda de archivos CAD

Cuando hayamos elegido un archivo (por ejemplo Base inferior.stl), la geometría aparecerá en nuestra estación, donde podremos trabajar con ella.

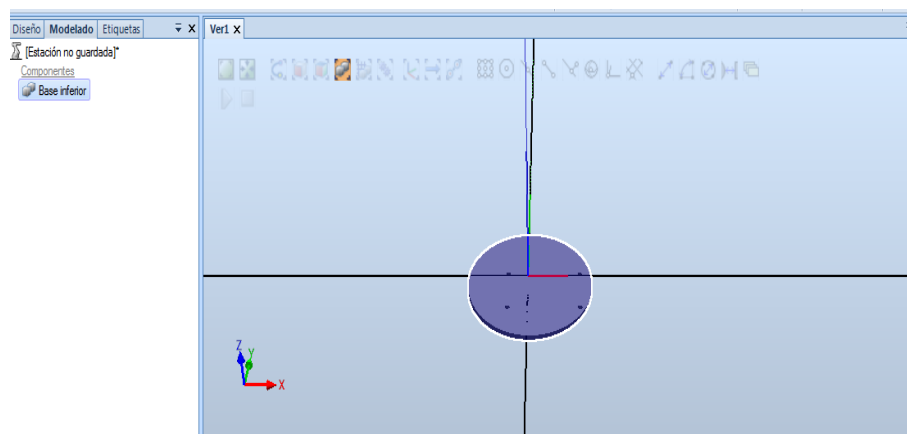


Figura 4-5. Geometría cargada en la estación

Una vez que hayamos importado todas las geometrías podremos tener nuestro brazo completo y comenzar a trabajar con él:

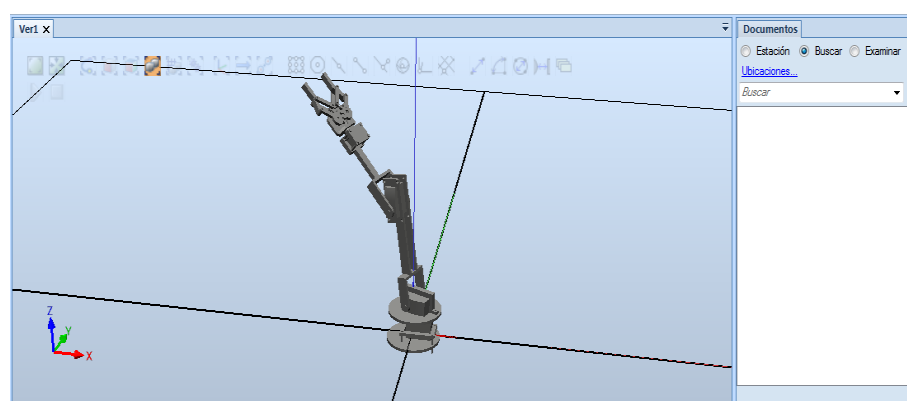


Figura 4-6. Brazo completo

4.1.3 Articulaciones y diseño de mecanismos

El siguiente paso que tenemos que dar es crear los mecanismos de las distintas articulaciones, con los que definiremos el movimiento de cada una de ellas.

4.1.3.1 Base

Para crear el mecanismo de la base vamos a utilizar los siguientes archivos CAD: Base inferior, Base superior, Soporte y Servo, quedando de la siguiente forma:

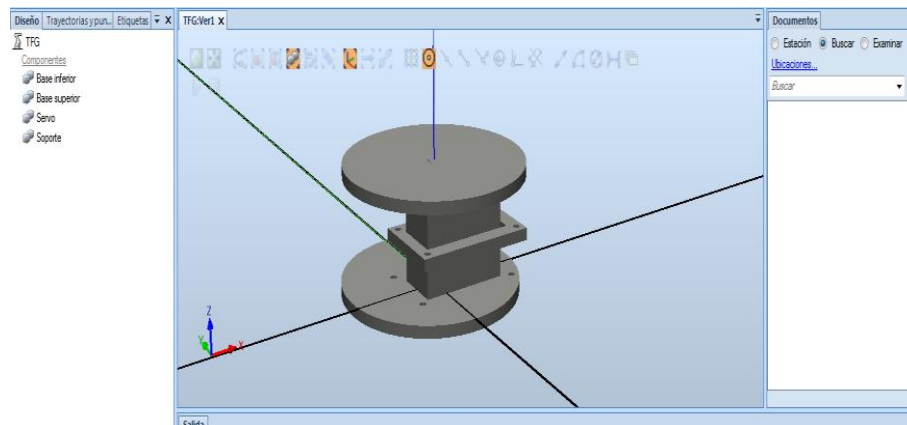


Figura 4-7. Base

Ahora debemos irnos a la pestaña “Modelado” y hacer click en “Crear mecanismo”:

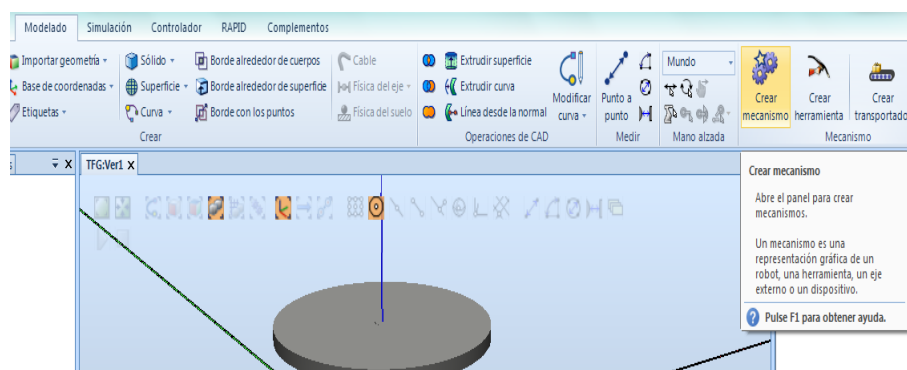


Figura 4-8. Crear mecanismo

Una vez que seleccionemos esta opción nos aparecerá el siguiente menú:

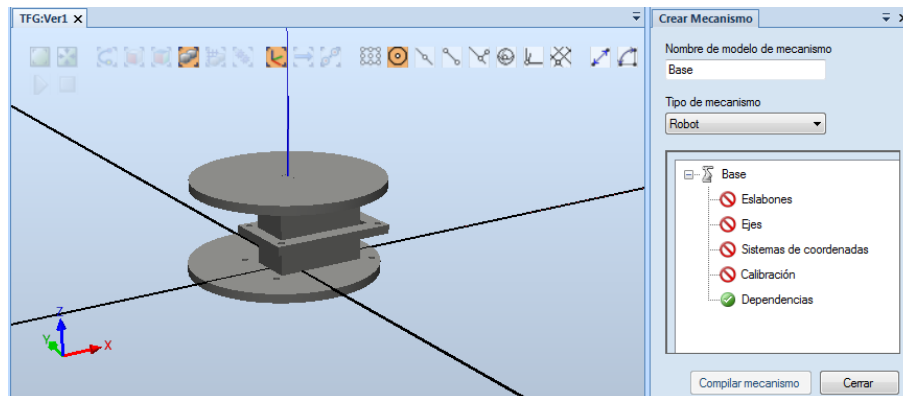


Figura 4-9. Menú de creación de mecanismo

En el menú vemos una serie de elementos que debemos definir: eslabones, ejes, sistemas de coordenadas y calibración. Vamos a comenzar definiendo los eslabones, haciendo click en “Eslabones” y veremos lo siguiente:

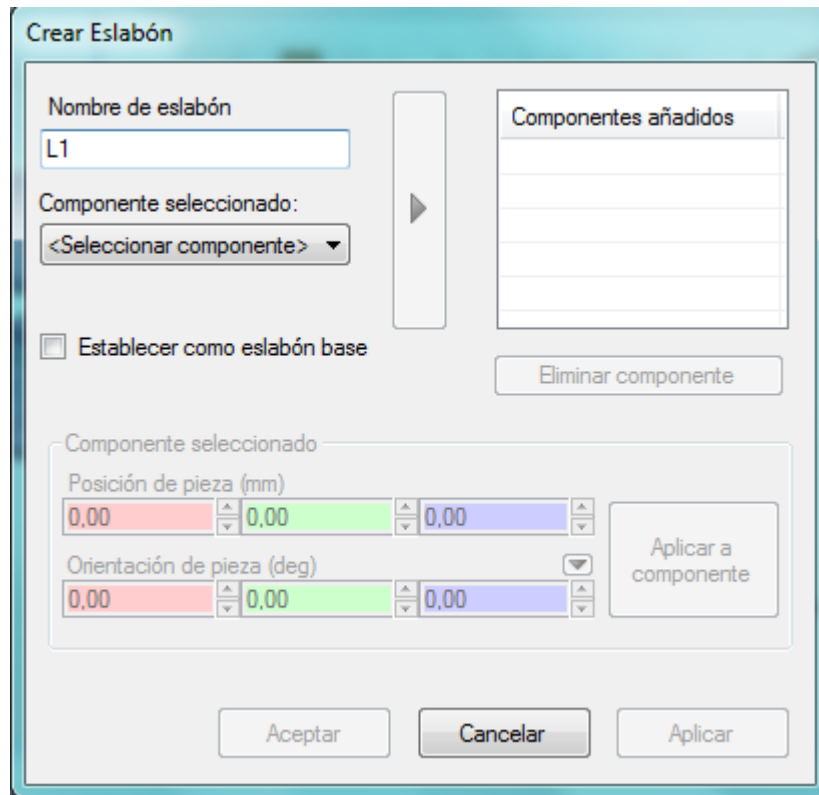


Figura 4-10. Ventana de creación de eslabón

En esta ventana debemos elegir un componente, ponerle un nombre y añadirlo como eslabón. También

tenemos la opción de elegir uno de los componentes como eslabón base, que en este caso será Base inferior. Una vez hayamos creado los eslabones veremos lo siguiente:

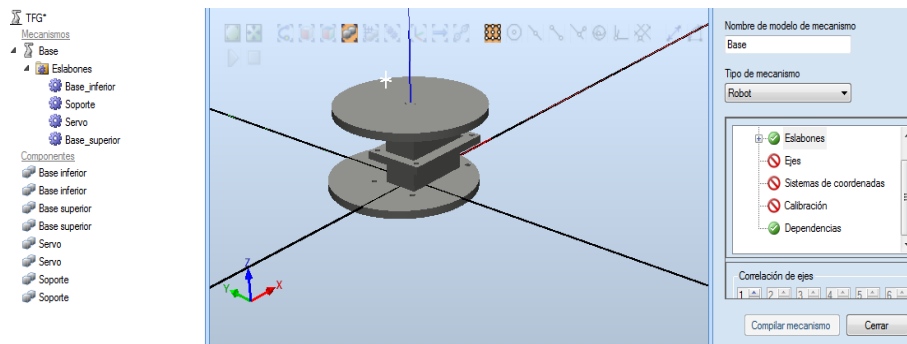


Figura 4-11. Creación de mecanismos: eslabones creados

Como podemos ver, la pestaña “Eslabones” aparece validada en el menú de creación de mecanismo, así como también aparecen los eslabones creados a la izquierda, dentro del título “Mecanismos”. Sigamos ahora con la definición de ejes. Si hacemos click en “Ejes” aparecerá la siguiente ventana:

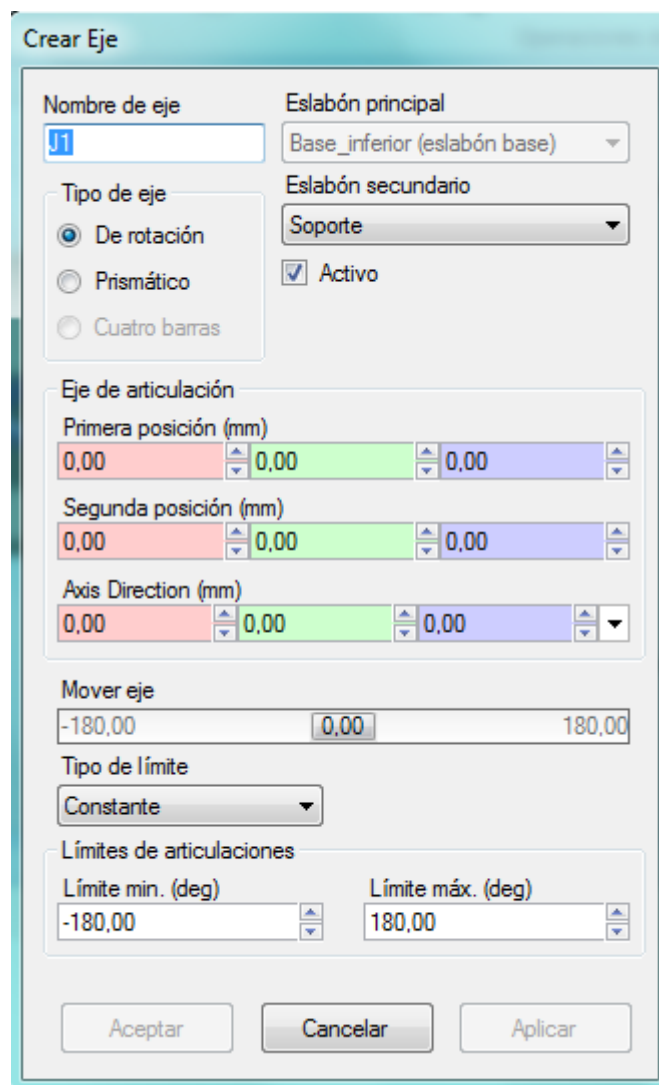


Figura 4-12. Ventana de creación de ejes

Vemos en esta ventana que podemos nombrar al eje como queramos, así como debemos escoger de qué tipo de eje se tratará (en nuestro caso, de rotación). Tras elegir los eslabones principal y secundario (base inferior y base superior, respectivamente), debemos escoger los puntos que definirán el eje sobre el cual rotará la base superior. Cuando lo hayamos elegido, podemos comprobar el movimiento en torno a ese eje mediante la barra de la función “Mover eje” que aparece un poco más abajo. Una vez hecho todo esto, pulsamos aplicar y nuestro eje habrá sido creado. Podemos ver a continuación lo siguiente:

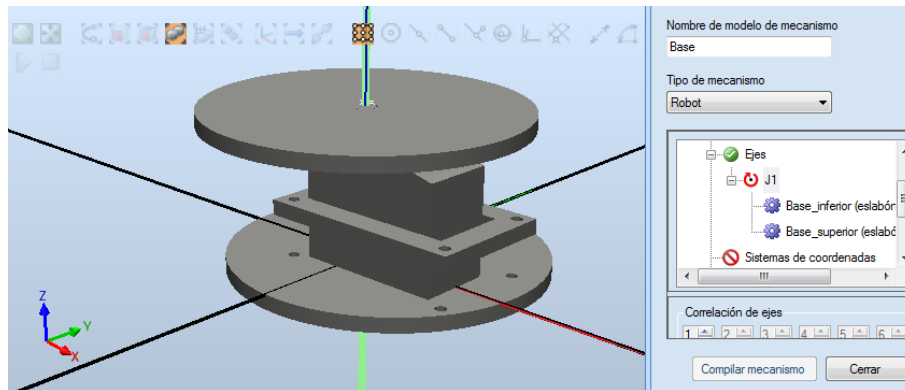


Figura 4-13. Creación de mecanismos: eje creado

Es fácil observar que la pestaña “Ejes” ha sido validada en el menú de creación del mecanismo, y en su interior aparece el eje creado, llamado J1, así como los eslabones involucrados. Además, observando la figura podemos ver el eje, representado por un haz verde. El siguiente paso que debemos dar es definir un sistema de coordenadas para nuestro mecanismo. Si hacemos click en “Base de coordenadas” aparecerá la siguiente ventana:

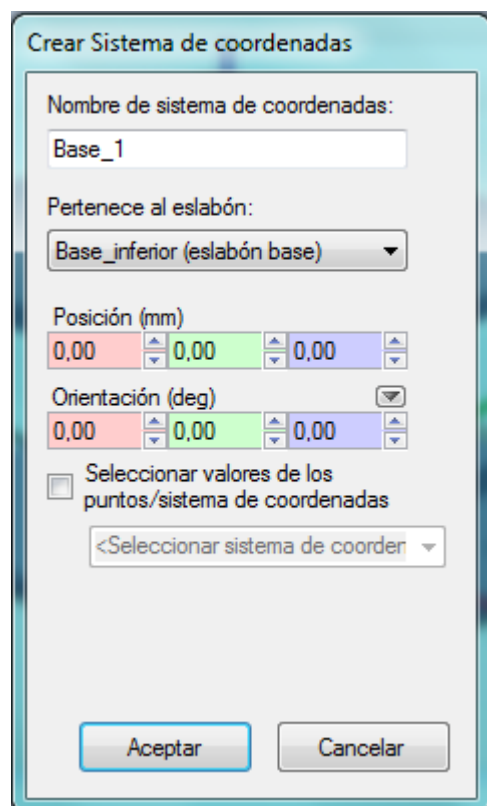


Figura 4-14. Ventana de creación de sistema de coordenadas

En esta ventana debemos darle un nombre a nuestro sistema de coordenadas, seleccionar al eslabón al que pertenece y definir su posición. Una vez hecho pulsamos aceptar y podremos ver lo siguiente en la ventana de

creación de mecanismos:

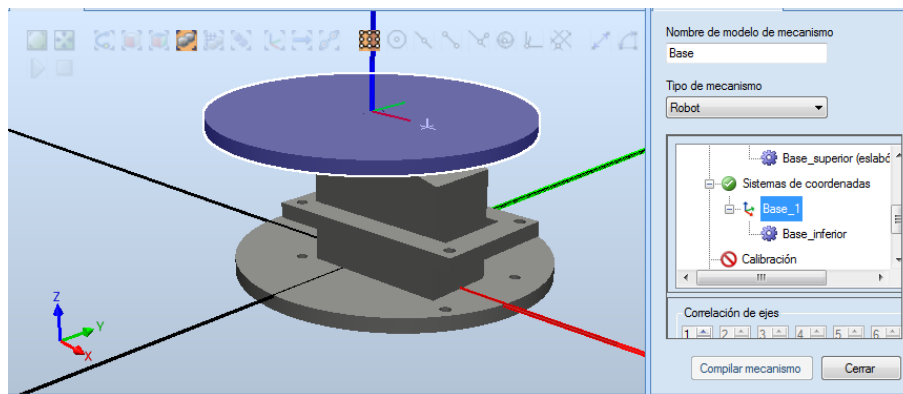


Figura 4-15. Creación de mecanismos: sistema de coordenadas creado

Como viene siendo habitual, la pestaña “Sistema de coordenadas” aparece validada, pudiendo ver creado el sistema de coordenadas, llamado Base_1, así como el eslabón al que pertenece. Vayamos ahora con la siguiente pestaña: “Calibración”. Si hacemos click en ella:



Figura 4-16. Ventana de creación de calibración

En esta ventana debemos seleccionar un eje para efectuar la calibración. Si desplegamos la pestaña nos aparecerán los ejes que hemos creado. En este caso solo hemos creado un eje, al que llamamos J1. Si lo seleccionamos y damos a aceptar, ya estará creada la calibración. Volviendo a la ventana de creación de mecanismos tenemos:

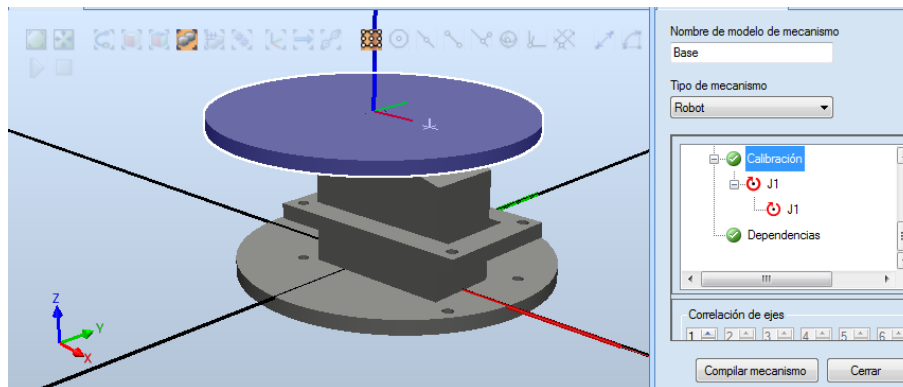


Figura 4-17. Creación de mecanismos: calibración creada

La pestaña de calibración ha sido validada, apareciendo dentro de ella el eje que se ha elegido para la calibración. Ya sólo quedaría la pestaña “Dependencias”, que está validada automáticamente ya que sólo hay un eje creado, por lo tanto no hay ninguna dependencia entre ejes.

Ahora que tenemos validadas todas las pestañas, podemos hacer click en compilar mecanismo, con lo que será creado. En el menú de la izquierda aparecerá nuestro mecanismo (al que llamamos Base). Si hacemos click derecho sobre él podemos elegir la opción “Movimiento de ejes de mecanismo”, como podemos ver en la siguiente imagen:

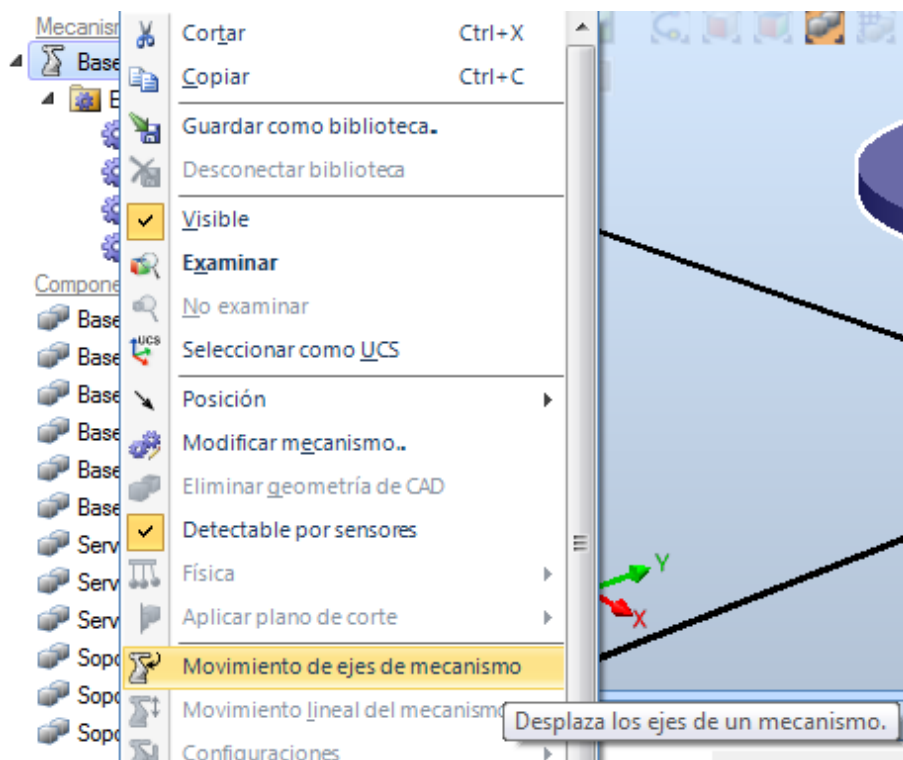


Figura 4-18. Selección de movimiento de ejes de mecanismo

Pulsando esta opción nos encontramos con lo siguiente:

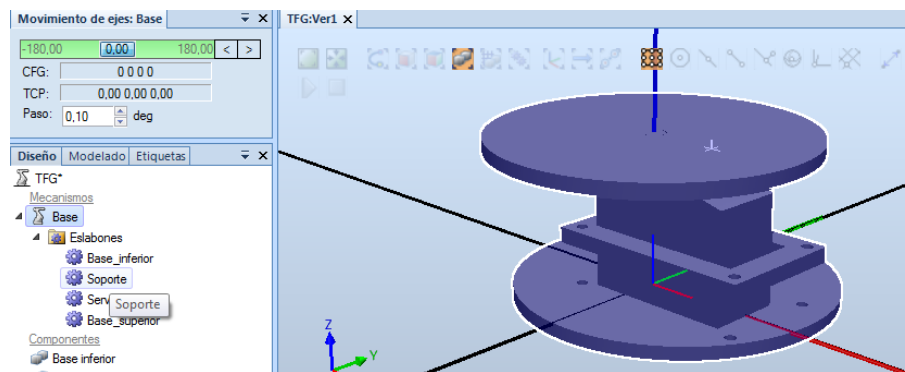


Figura 4-19. Movimiento de ejes de mecanismo

Esto nos permite mover la base superior en torno al eje y observar como se produce el movimiento. Tenemos una barra con un fondo verde, cuyos extremos son -180 y 180. Tenemos dos opciones, o bien movemos el rectángulo por la barra, donde se le irá asignando un valor al ángulo de giro, o bien podemos hacer click en el rectángulo y escribir directamente el valor que queremos que tenga el ángulo.

4.1.3.2 Primera articulación

Vayamos ahora a definir el mecanismo de la primera articulación, la cual se compone de las siguientes piezas: “Servo”, “Sujeta-servo” y “Arti”. Antes de montar la articulación y crear el mecanismo correspondiente hemos de hacer lo siguiente: debemos conectar el servo de la articulación a la base, ya que así el servo quedará anclado a ella y cuando usemos el mecanismo para mover la base superior, el servo se moverá con ella.

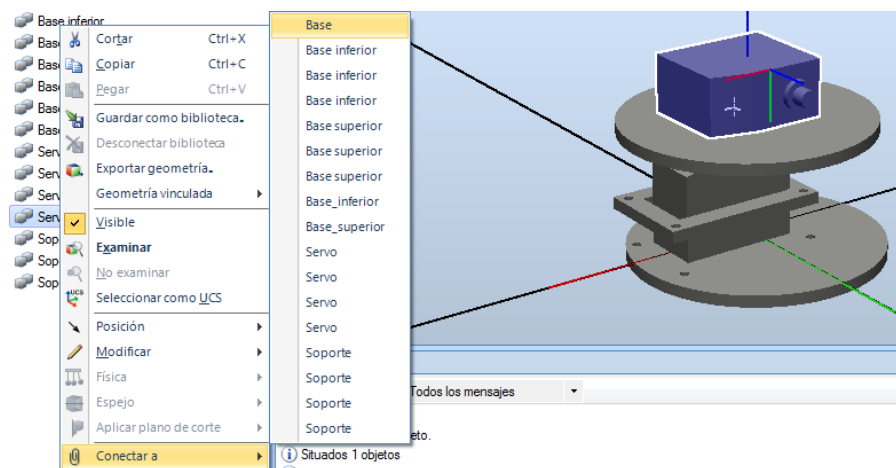


Figura 4-20. Conexión de piezas

Una vez hecho esto, debemos colocar las dos piezas restantes, para poder formar la articulación, que quedaría de la siguiente manera:

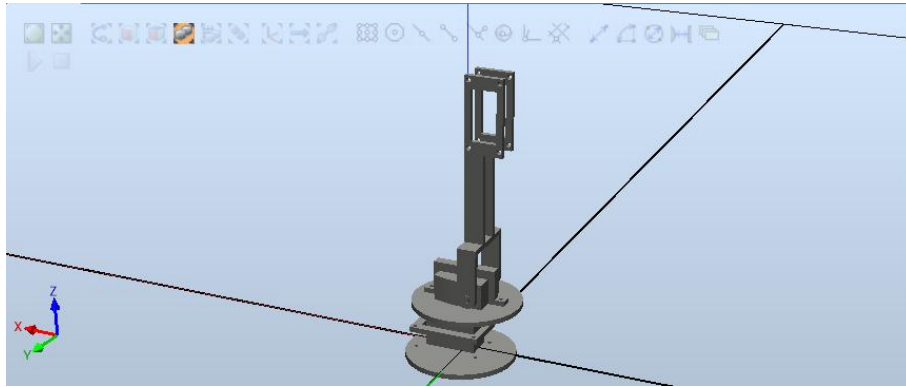


Figura 4-21. Primera articulación

Ahora debemos crear un mecanismo, tal y como hicimos con la base. La manera de proceder es idéntica a como la hicimos con la base, así que nos detendremos menos. Recordemos que lo primero era crear los eslabones: elegiremos como eslabón base el servo, que está conectado a la base superior. Los otros dos eslabones serán el soporte que sujeta al servo y la articulación.

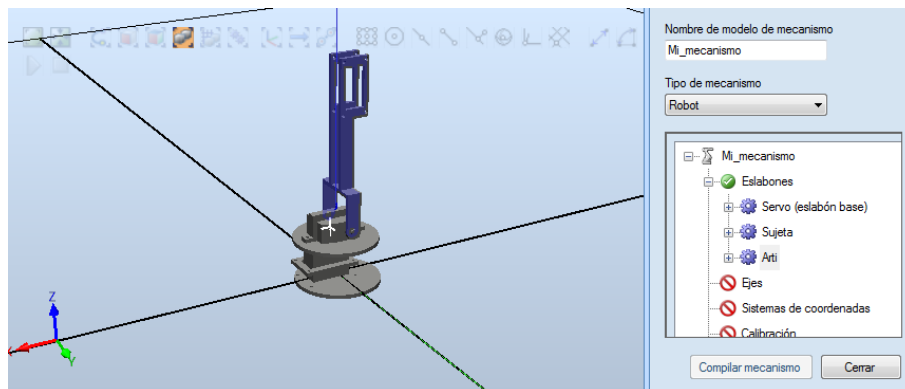


Figura 4-22. Primera articulación: eslabones creados

El siguiente paso era definir el eje del mecanismo. En este caso se define como eslabón principal el servo y como eslabón secundario Arti. Se define el eje de esta manera:

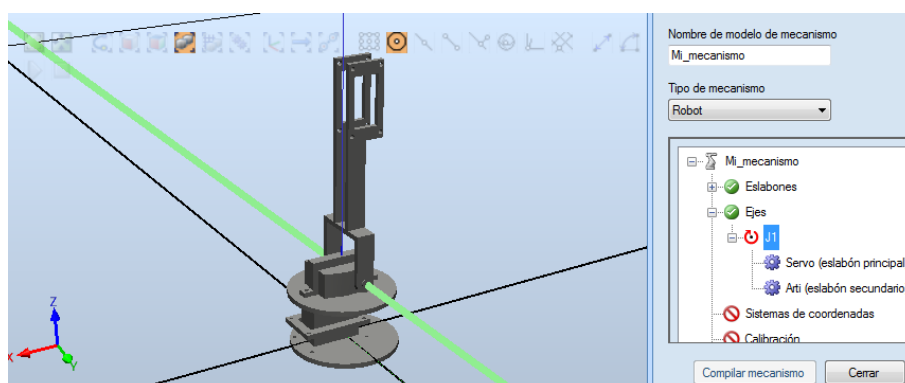


Figura 4-23. Primera articulación: eje de giro

Vemos que será Arti la que gire en torno a ese eje. El rango del giro son -90° y 90° , para ser coherente con el giro que proporciona el servo y para que Arti no atravesase la base.

Después de esto, definimos un sistema de coordenadas perteneciente a la pieza Arti y calibramos, tras lo cual

ya tendremos nuestro mecanismo creado. Como hicimos con el anterior mecanismo, comprobamos que usando la selección de movimiento de ejes de mecanismo la articulación se mueve como esperábamos. Además, si vinculamos este mecanismo al de la base, cuando la base superior efectúe un giro, la articulación girará con ella.

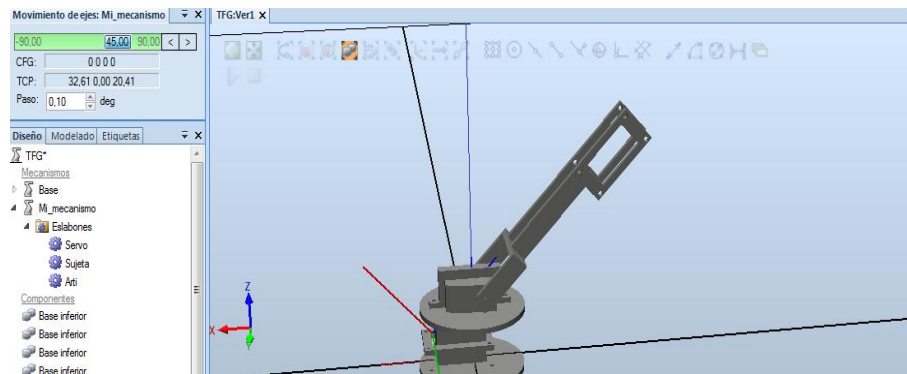


Figura 4-24. Articulación girada 45°

4.1.3.3 Segunda articulación

Pasamos ahora a definir el mecanismo de la segunda articulación, la cual se compone de las siguientes piezas: “Arti-dos”, “Servo”, “Servo pequeño” y “Tab-servo”. Una vez colocados y montada la articulación sobre el resto del brazo, quedaría de la siguiente forma:

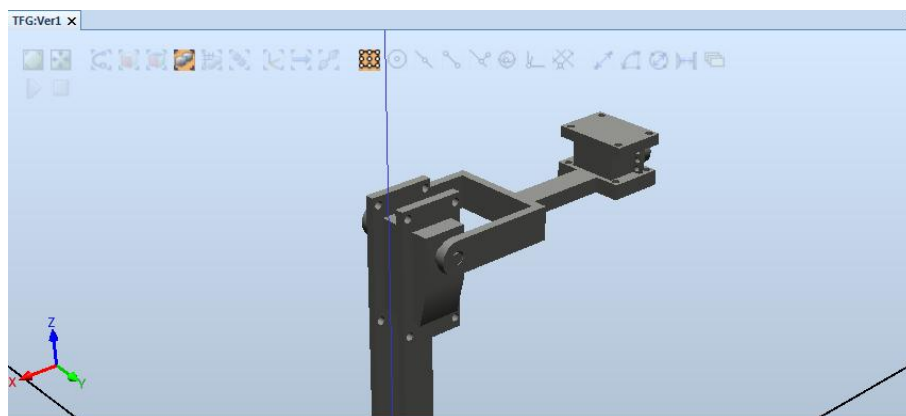


Figura 4-25. Segunda articulación

Procedemos ahora a crear el mecanismo de esta articulación, de similar manera a como creamos los anteriores. Lo primero que debemos hacer es seleccionar los eslabones que formarán esta articulación: elegiremos como eslabón base el servo, que estará conectado a la anterior articulación. Cabe destacar que, en este caso, dentro de un mismo eslabón agruparemos el resto de piezas. Llamaremos “Segunda articulación” a este eslabón. El motivo de trabajar así es el siguiente: al trabajar únicamente con dos eslabones, cuando definamos el eje de giro conseguiremos que todas las piezas que forman el segundo eslabón se muevan coherentemente la una con la otra. De no hacerlo así, cada una tendría un giro por separado, lo cual, aparte de no recrear el comportamiento real del brazo, puede darnos problemas a la hora de seguir trabajando.

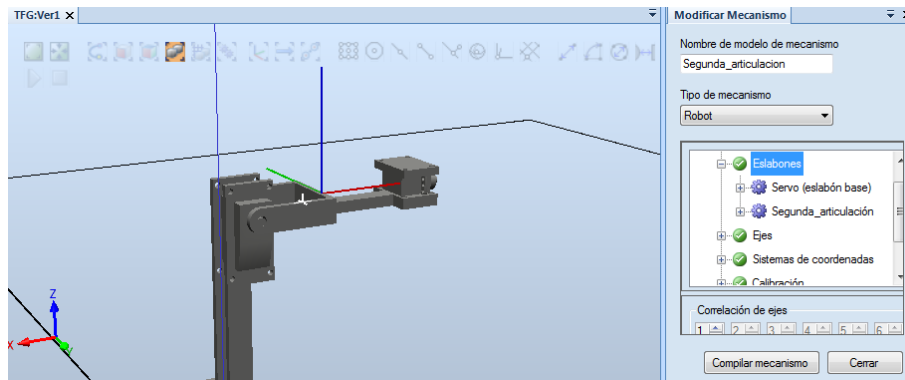


Figura 4-26. Segunda articulación: eslabones creados

El siguiente paso es definir el eje de giro, para el cual se elige como eslabón principal el servo y como eslabón secundario Segunda_articulación, acotando los valores del ángulo entre -90° y 90° . Se define el eje de esta manera:

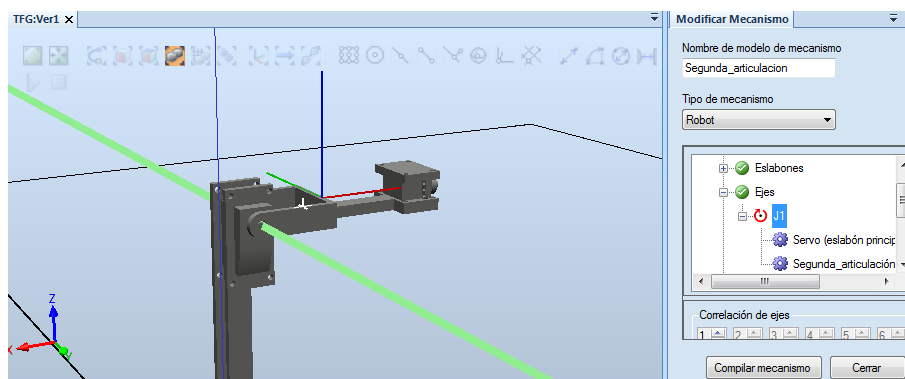
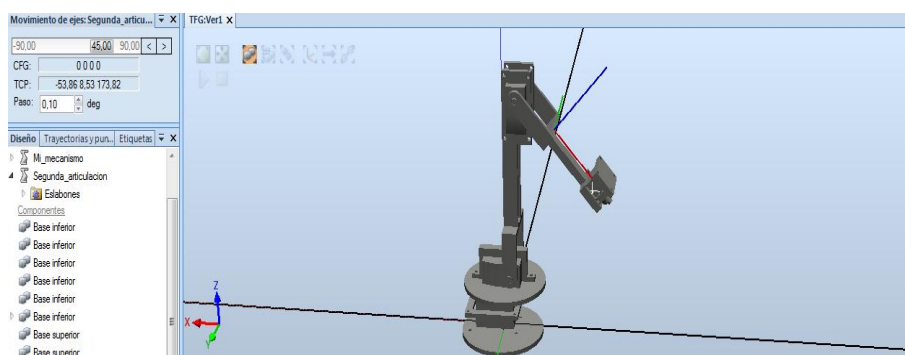


Figura 4-27. Segunda articulación: eje de giro

Se puede ver que serán las piezas pertenecientes a Segunda_articulación quien gire alrededor de ese eje.

Ahora debemos asignar un sistema de coordenadas perteneciente al eslabón Segunda_articulación y calibramos, tras lo cual ya tendremos el mecanismo creado. Pasamos ahora a comprobar que el mecanismo se mueve como esperábamos. Además, si conectamos este mecanismo al anterior, girará si lo hace éste.

Figura 4-28. Articulación girada 45°

4.1.3.4 Garra

Vamos ahora con la parte final del brazo: la garra. Tiene un gran número de piezas comparada con el resto de articulaciones, y una vez montada quedaría de la siguiente manera:

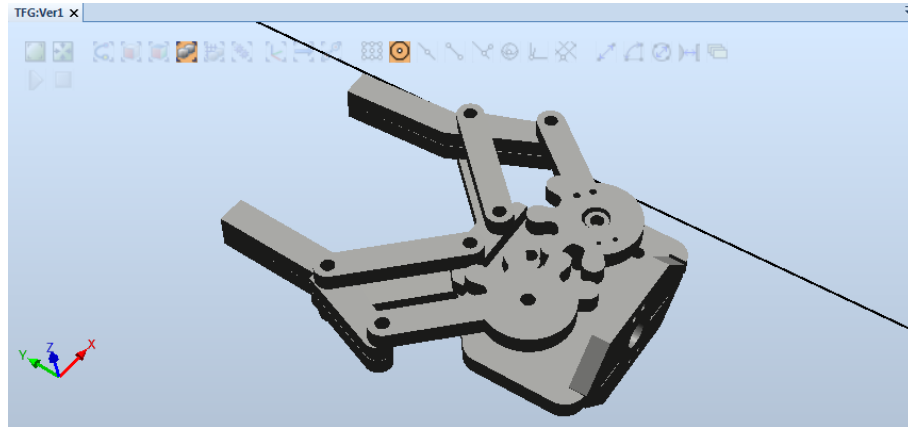


Figura 4-29. Garra

Hecho esto, el siguiente paso es anclarla al brazo y crear los mecanismos correspondientes. En este caso serán dos, uno que se encargue del giro de la garra y otro del agarre de la misma. Comenzaremos diseñando el mecanismo que se encarga del giro de la garra, al que llamaremos Giro_garra:

•Giro_garra

El primer paso es, como siempre, crear y definir los eslabones. En este caso elegimos como eslabón base el servo de la segunda articulación, que será el que se ocupe de hacer girar la garra. El otro eslabón, al que llamaremos Garra, se trata del cuerpo de la garra, y es el conjunto de piezas que se muestra en la imagen:

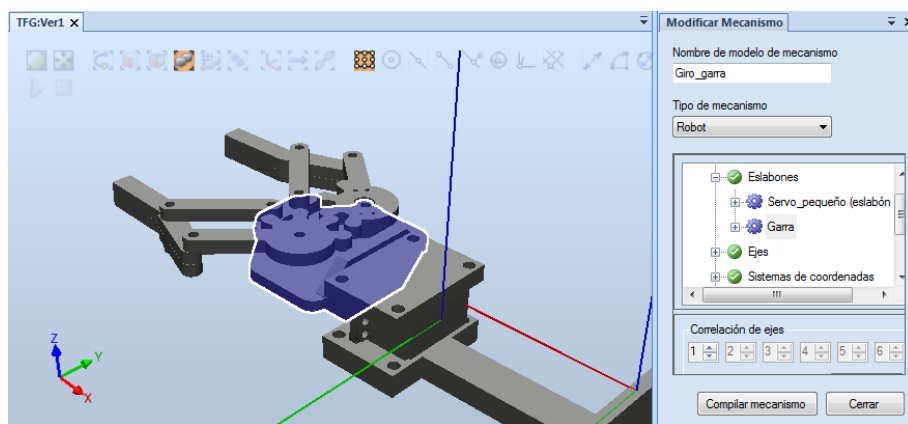


Figura 4-30. Giro_garra: eslabones creados

Una vez hecho esto, debemos crear un eje de giro, que pasará por el servo y atravesará la garra, de la manera en la que se ve en la imagen. Como hicimos con los anteriores, se limita el giro entre -90° y 90° .

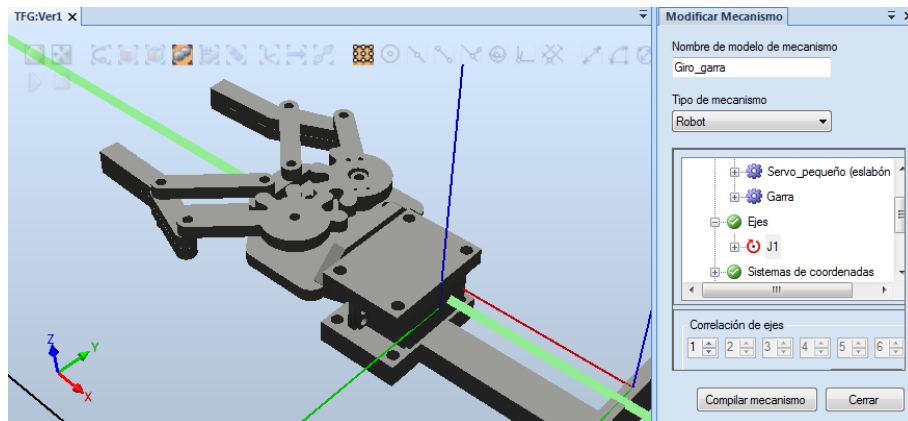


Figura 4-31. Giro_garra: eje creado

Puede verse que la garrá girará en torno a ese eje. Lo siguiente es asignar un sistema de coordenadas al mecanismo, que definimos en el servo que utilizamos de eslabón base. Tras esto, calibramos y compilamos el mecanismo. Conectando el resto de la garra al mecanismo, y conectando éste a la articulación anterior, conseguimos poder girarlo y que éste siga los movimientos del resto de mecanismo.

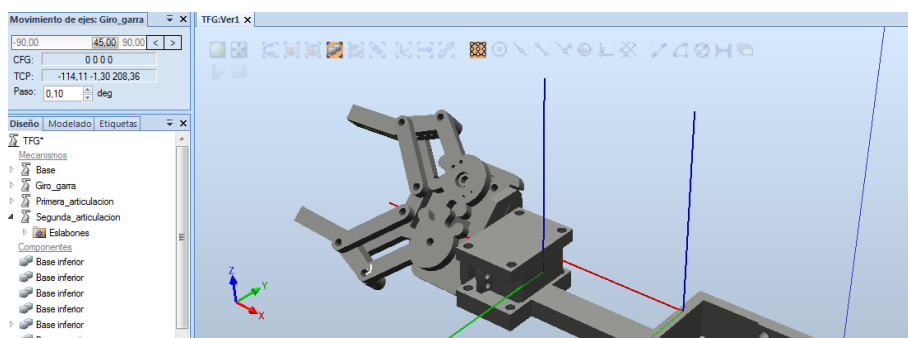


Figura 4-32. Garra girada 45°

•Cierra_garra

Vamos ahora con el mecanismo encargado de abrir y cerrar la garra, al que llamaremos Cierra_garra. Vamos a comenzar creando los eslabones. Vamos a tomar como eslabón base el servo pequeño que está colocado debajo del cuerpo de la garra. Aparte, crearemos dos eslabones más: Garra_derecha y Garra_izquierda. En la imagen puede verse, por ejemplo, que piezas pertenecen a Garra_derecha (siendo Garra_izquierda su simétrica):

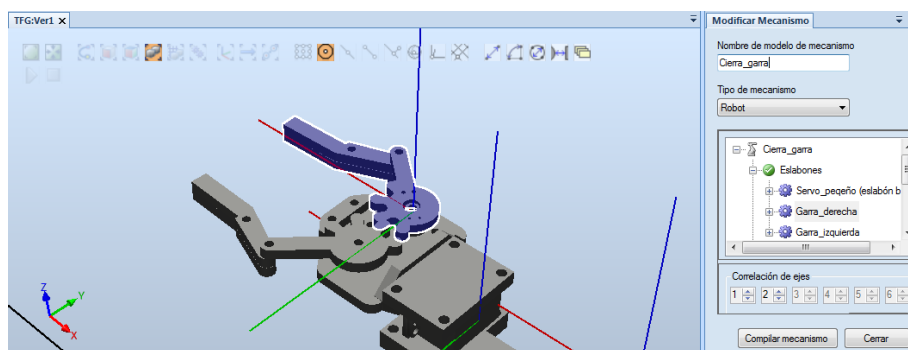


Figura 4-33. Cierra_garra: eslabones creados

Cabe destacar que se han eliminado las barras paralelas de la garra, y se ha hecho por el siguiente motivo: su movimiento era difícil de coordinar con el cierre y/o apertura de la garra, por tanto se ha decidido prescindir de ellas y hacer el movimiento más simple, entendiendo que el objetivo y el resultado final no se ven especialmente afectados.

Pasamos ahora a definir los ejes del mecanismo. En este mecanismo tendremos que definir dos ejes distintos, uno para Garra_derecha y otro para Garra_izquierda. Son los siguientes:

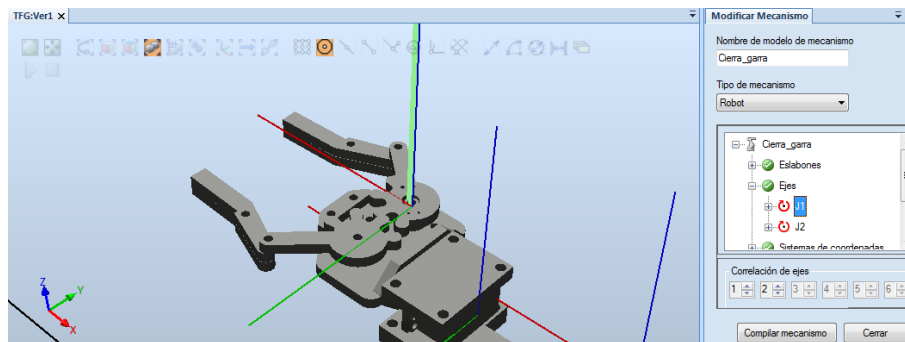


Figura 4-34. Cierra_garra: primer eje (J1) creado

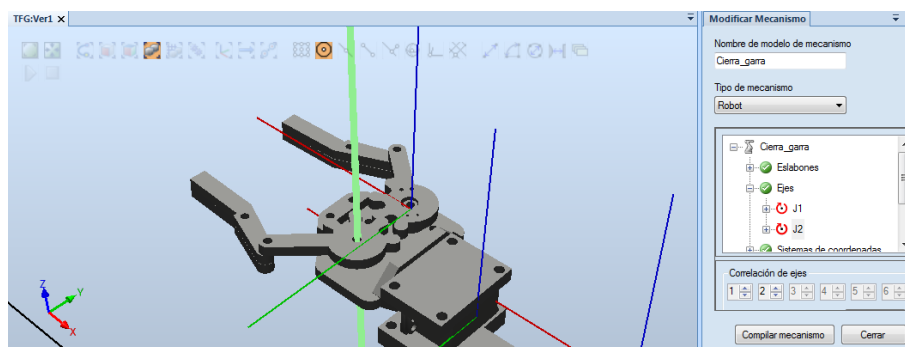
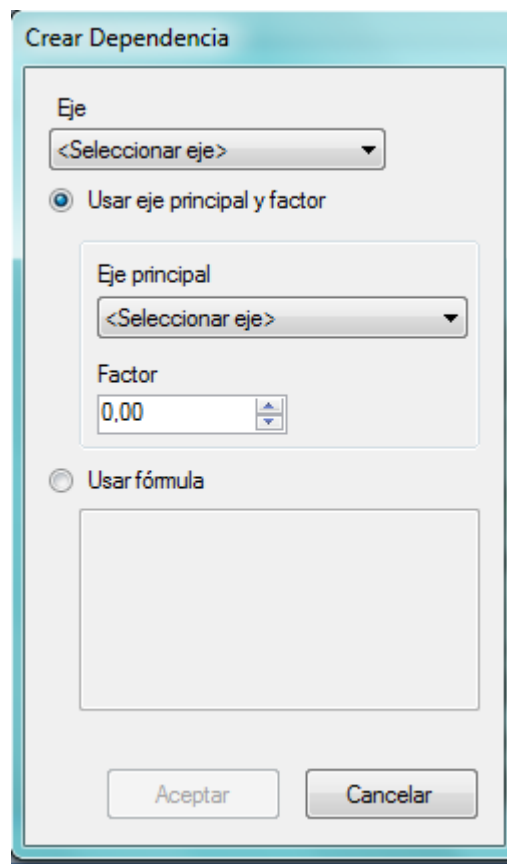


Figura 4-35. Cierra_garra: segundo eje (J2) creado

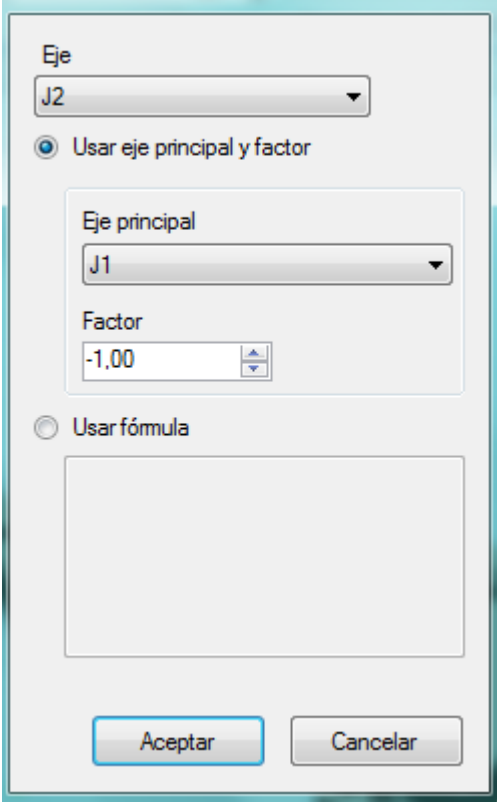
Esta vez vamos a definir los límites de los ejes en -10° y 10° , ya que parece un movimiento verosímil, y reproduce bien la apertura y el cierre. De las imágenes puede intuirse que Garra_derecha y Garra_izquierda deberán hacer sus giros en sentido contrario el uno del otro, ya que si no es así no conseguiremos el movimiento deseado y, por tanto, la garra no podrá cumplir su función. Esto lo conseguiremos usando la pestaña “Dependencias”, la cual no hemos tenido que usar hasta ahora porque no teníamos más de un eje por articulación. Tras haber definido un sistema de coordenadas y haber calibrado, vamos a crear la relación existente entre ambos ejes. Cuando queramos crear una dependencia nos aparecerá la siguiente ventana:



The image shows a dialog box titled "Crear Dependencia" (Create Dependency). It contains two main sections. The first section, "Usar eje principal y factor" (Use main axis and factor), is selected with a radio button. It includes a dropdown menu for "Eje" (Axis) with the text "<Seleccionar eje>", a sub-section for "Eje principal" (Main axis) with another "<Seleccionar eje>" dropdown, and a "Factor" (Factor) input field with the value "0,00" and a spinner control. The second section, "Usar fórmula" (Use formula), is unselected. It features a large empty text area for entering a formula. At the bottom of the dialog are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Figura 4-36. Ventana de creación de dependencia

En la primera pestaña debemos seleccionar el eje dependiente, que en nuestro caso será J2, el correspondiente a Garra_izquierda. El siguiente paso es elegir el eje principal, que será J1 (correspondiente a Garra_derecha) y, por último, debemos elegir el factor que los relacione. En nuestro caso será -1, ya que deben efectuar el mismo giro pero de sentido contrario. Quedaría así:



The image shows a dialog box titled 'Eje' (Axis). It has two main sections. The first section is for 'Usar eje principal y factor' (Use main axis and factor), which is selected with a radio button. It contains a dropdown for 'Eje' (Axis) set to 'J2', a dropdown for 'Eje principal' (Main axis) set to 'J1', and a numeric input for 'Factor' (Factor) set to '-1,00'. The second section is for 'Usar fórmula' (Use formula), which is unselected. It contains a large empty text area. At the bottom are two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel).

Figura 4-37. Creación de dependencia

Cabe destacar lo siguiente: a efectos prácticos es indiferente cual es el eje principal y cual es el dependiente, funcionaría de igual manera intercambiando J1 y J2. El motivo de haber elegido J1 como principal no es otro que el de tratar de respetar la realidad, ya que es la garra derecha la que es movida por el servo; y es ella, a su vez, quien imprime movimiento a la garra izquierda.

Ya podemos compilar el mecanismo, y debemos conectarlo a Garra_giro para que sea capaz de seguir los movimientos del resto del brazo. Usando la función de movimiento de ejes del mecanismo podemos observar que funciona correctamente:

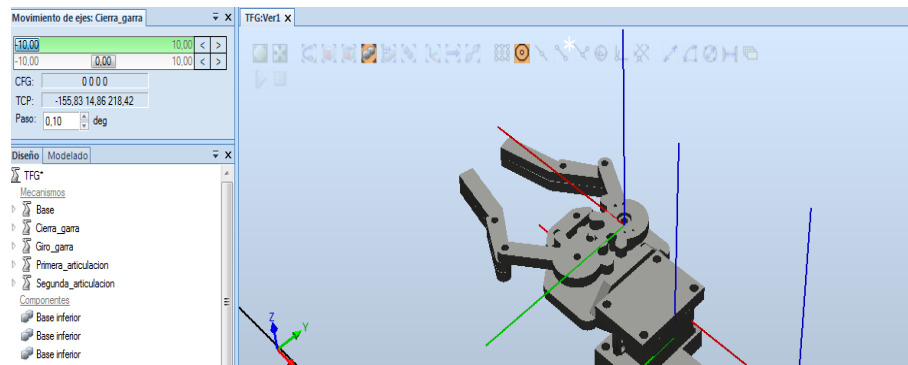


Figura 4-38. Garra cerrada

Ahora que hemos construido el brazo y hemos diseñado todos sus mecanismos, vamos ahora a explicar y trabajar con los Smart Components, aplicándolos a nuestra garra.

4.2 Smart Components

Los Smart Components o Componentes Inteligentes son elementos asociados a los sólidos, piezas o robots de la estación, los cuales tienen un comportamiento controlado por señales y propiedades del sistema. Si nos vamos a la pestaña “Modelado” veremos que tenemos la opción “Componente inteligente”:

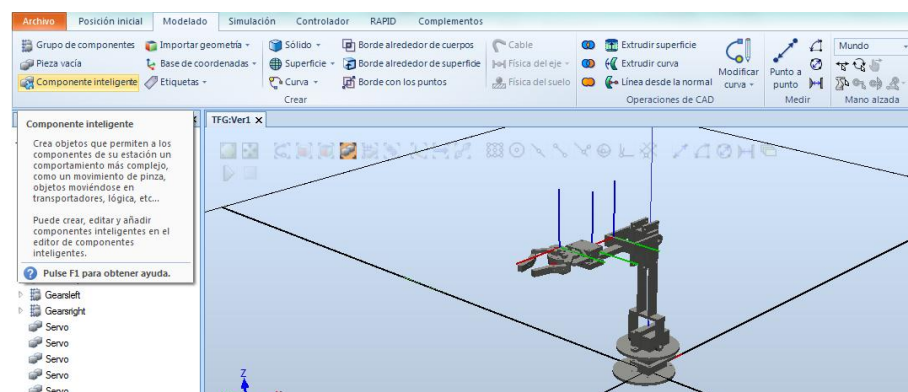


Figura 4-39. Creación de Smart Components

Una vez que seleccionemos la opción nos aparece la siguiente ventana:



Figura 4-40. Ventana de creación de Smart Components

Una vez en esta ventana, debemos hacer click en añadir componente, de entre los cuales podemos elegir de seis tipos distintos:

-Señales y propiedades: aquí podemos seleccionar puertas lógicas, expresiones matemáticas, contadores, comparadores, etc., con los que podremos modificar o actuar sobre una señal o propiedad del sistema.

-Primitivos paramétricos: aquí tenemos diversos componentes con los que podemos crear sólidos, líneas o crear copias de otro componente gráfico.

-Sensores: en este apartado podemos encontrar los distintos sensores que podemos utilizar en nuestra estación, tales como sensor de colisión, de línea, volumétrico, etc.

-Acciones: aquí tenemos componentes que nos permitirán distintas acciones como conectar/desconectar un objeto de otro, hacer un objeto visible o invisible, etc.

-Manipuladores: aquí tenemos componentes con los que podremos efectuar distintas acciones, tales como mover objetos de manera lineal, rotarlos alrededor de un eje, moverse a lo largo de una curva o posicionarlo en un lugar determinado. También permite mover los ejes de un mecanismo.

-Otros: es la última categoría, y en ella podemos encontrar algunos Smart Components que nos permitiría representar una cola de objetos, generación de números aleatorios, reproducir sonidos, etc.

Lo que vamos a buscar mediante los Smart Components es poder abrir y cerrar la garra mediante el uso de una señal. Para ello debemos definir una serie de componentes subordinados al Smart Component que llamaremos SC_Garra, de manera que podamos acabar cumpliendo la función deseada. Para ello vamos a ver los componentes que necesitamos uno a uno, para más tarde explicar como se conectan entre ellos para conseguir nuestro objetivo.

•Cierra_Pinza

Lo primero que haremos será añadir a SC_Pinza un componente “JointMover”, que nos permitirá mover los ejes de un mecanismo determinado. Una vez lo hayamos añadido, si nos vamos a la pestaña “Diseño”, que puede verse en la imagen 4-38, nos iremos a la siguiente ventana:

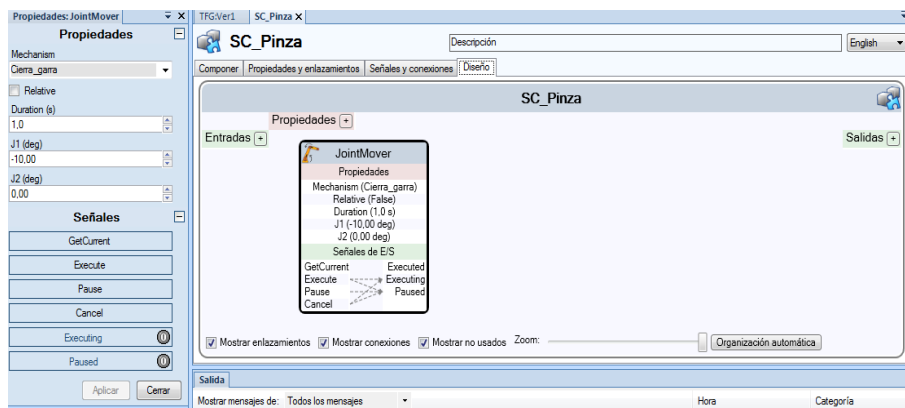


Figura 4-41. Definición de Cierra_Pinza

A la izquierda tenemos las propiedades del componente, que hemos de definir. En primer lugar, designamos “Cierra_garra” como el mecanismo sobre el que vamos a actuar con este componente. Tras esto, le asignamos una duración de un segundo y definimos una actuación sobre el eje J1 de -10° que, si recordamos, fue el valor límite que definimos en el mecanismo para indicar que la garra estaba cerrada (cabe recordar también que, debido a la dependencia que creamos en el mecanismo, si actuamos sobre J1 también lo estaremos haciendo sobre J2).

•Abre_Pinza

A continuación añadiremos otro componente “JointMover”, en este caso su función será la opuesta al anterior: abrir la garra. Nuevamente nos vamos a la pestaña “Diseño” tras haber añadido el componente y tendremos lo siguiente:

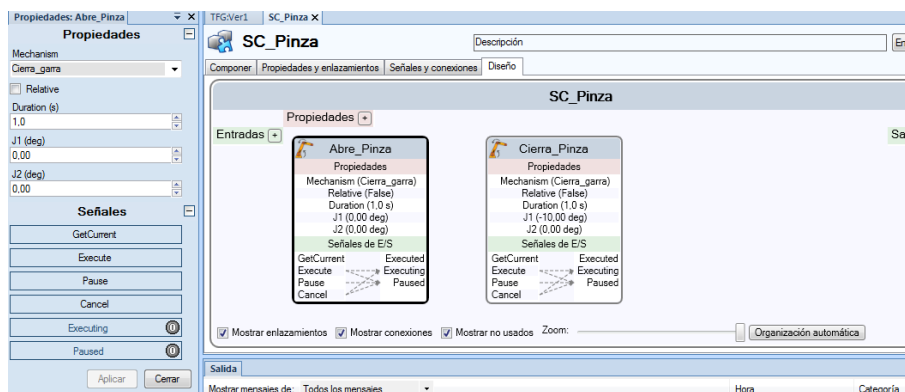


Figura 4-42. Definición de Abre_Pinza

Como hicimos anteriormente, definimos “Cierra_garra” como el mecanismo sobre el que actuará este componente. Le asignamos una duración de un segundo y definimos una actuación sobre el eje J1 de manera que su valor sea 0° , que es la posición inicial y en la que consideramos que la garra está abierta (puede llegar hasta el valor 10° , pero suponemos que no es necesario llegar al valor límite para que la garra se considere abierta).

•Sensor_pinza

El siguiente componente que añadiremos será un “PlaneSensor”, un sensor plano que llamaremos

“Sensor_pinza”. Colocaremos este sensor en la cara interna de la garra para que nos informe de si se está sujetando una pieza o no. Si ahora nos vamos a la pestaña “Diseño” tenemos lo siguiente:

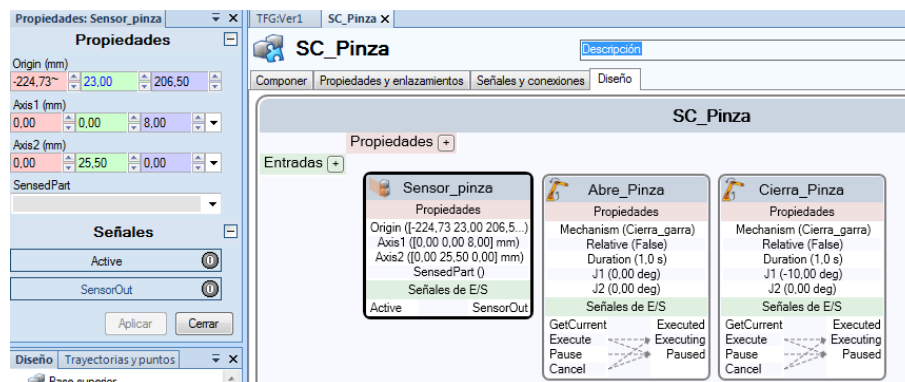


Figura 4-43. Definición de Sensor_pinza

En “Propiedades” vamos a definir una serie de cosas: en primer lugar, en “Origin” vamos a definir donde se encuentra el sensor. Por otro lado, mediante “Axis 1” y “Axis 2” vamos a definir la altura y anchura del sensor, respectivamente. Una vez definido, lo colocamos en la garra de la siguiente manera:

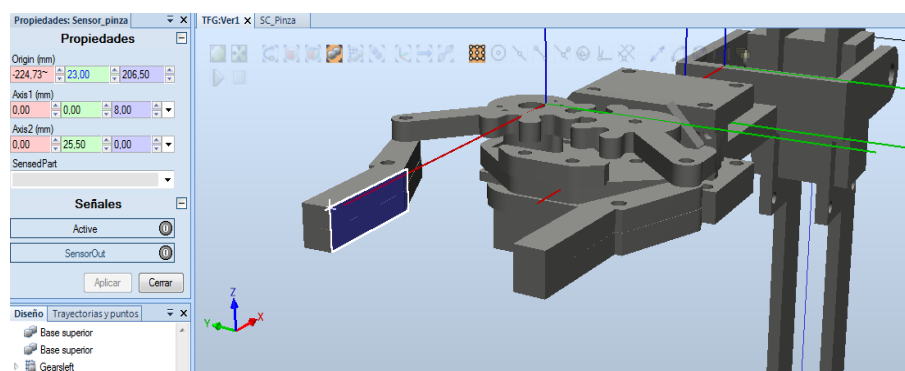


Figura 4-44. Ubicación de Sensor_pinza

•Attacher y Detacher

Estos componentes se usarán para las siguientes funciones: conectar un objeto a otro (Attacher) y desconectarlos (Detacher). Si nos vamos a la pestaña “Diseño” tenemos:

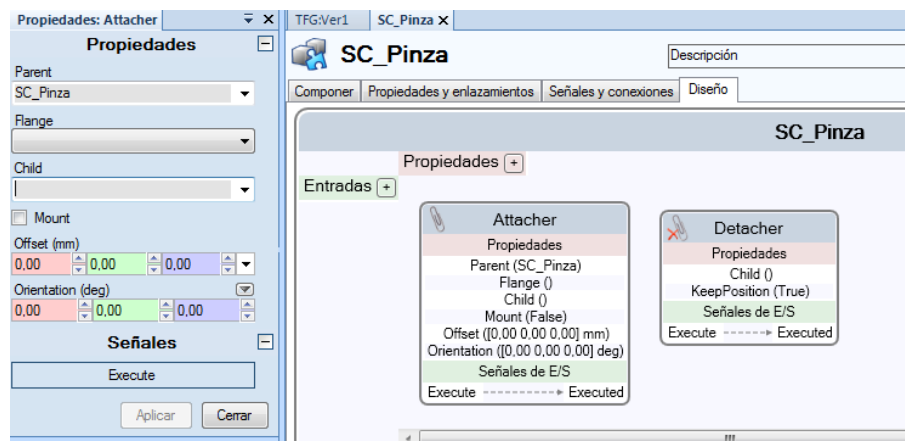


Figura 4-45. Definición de Attacher y Detacher

En las propiedades del componente Attacher, en Parent debemos elegir el objeto superior, aquel al que se vinculará el objeto que sea sujetado por la garra (Child). Nosotros elegimos como Parent el mecanismo “Cierra_garra”, que será al que se conecte el objeto. Por otro lado, en las propiedades de Detacher lo dejamos tal y como está, ya que no debemos definir nada de momento.

4.2.1 Señales, enlazamiento y diseños

El siguiente paso que daremos será el de definir las señales de entrada y salida. Para ello, debemos irnos a la pestaña “Diseño”. Una vez allí, debemos hacer click sobre “Entradas”, en la parte superior izquierda, y nos aparecerá la siguiente pestaña:

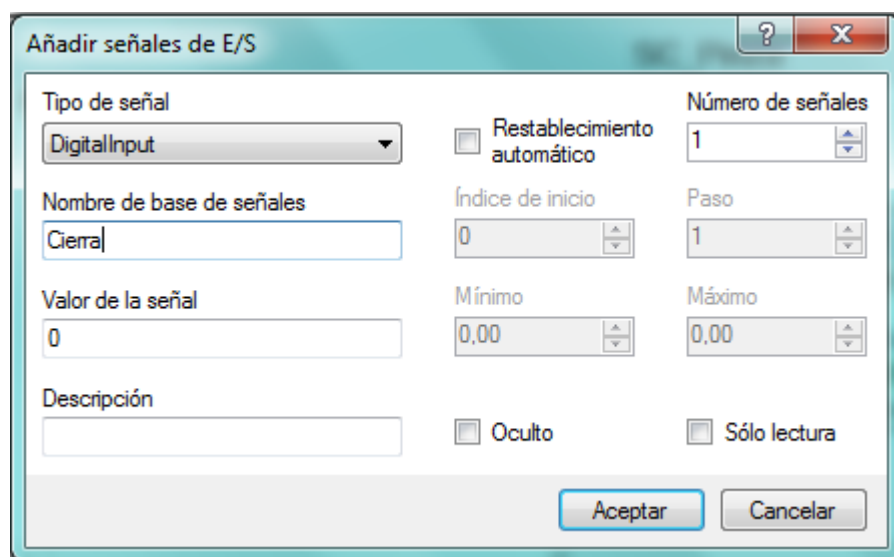


Figura 4-46. Ventana para añadir señales de entrada/salida

En esta ventana podemos seleccionar el tipo de señal, que en nuestro caso la dejaremos como entrada digital, y le daremos el nombre “Cierra”, así como le asignaremos valor cero. Esta señal será la que hará que la garra se cierre y pueda agarrar un objeto. De igual manera, en la pestaña si vamos al lado opuesto a “Entradas” (o sea, en la esquina superior derecha) podemos encontrar “Salidas”. Si lo pulsamos se nos abrirá una ventana idéntica a la mostrada en la figura 4-44, en la que definiremos una señal digital de salida que llamaremos “Cogida”. Esta señal nos indicará que una pieza ha sido agarrada (y esto será detectado por el sensor plano que

definimos en el punto anterior).

Una vez hecho esto, vamos a proceder al diseño de SC_Pinza usando todos los componentes que le hemos añadido, y tenemos el siguiente esquema:

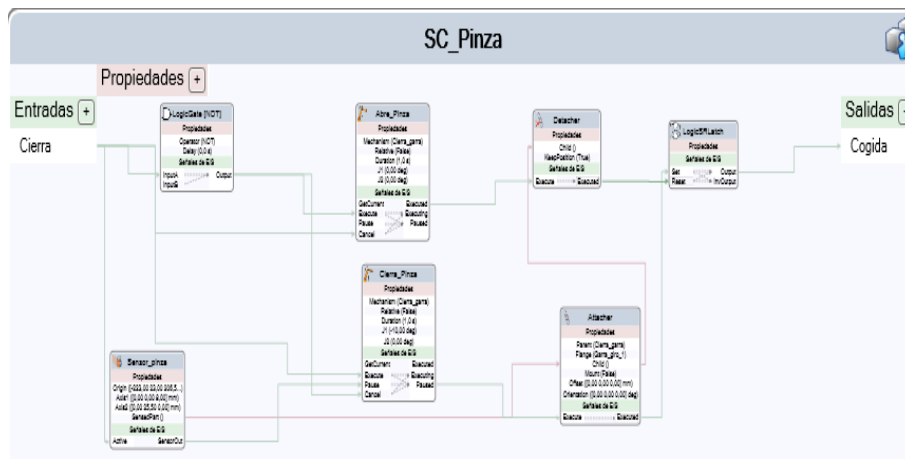


Figura 4-47. Diseño de SC_Pinza

En la imagen podemos ver que, al activar la señal Cierra ocurren dos cosas: se activa Sensor_pinza y se ejecuta Cierra_pinza. Además, mediante una puerta lógica NOT que hemos añadido, si la señal Cierra vale 0, se ejecutará Abre_Pinza. Si el sensor reconoce una pieza pausará Cierra_Pinza, y esto activará el Attacher, que conectará el objeto detectado por el sensor a la garra y activará la señal Cogida, mediante el bloque LogicSRLatch. Por otro lado, al ejecutarse Abre_Pinza se activa el Detacher y se desconectará el objeto que hubiese conectado a la garra (si hubiera alguno), lo que pondrá el valor de Cogida a 0, mediante el bloque LogicSRLatch.

Vamos a comprobar ahora si funciona. Si pulsamos sobre SC_Pinza nos aparecerá lo siguiente:

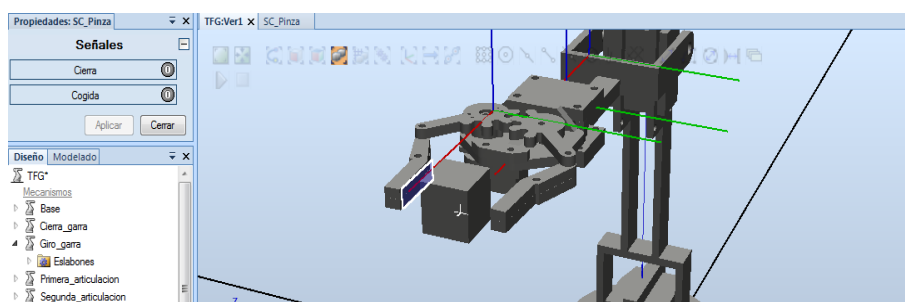


Figura 4-48. Propiedades de SC_Pinza: Cierra desactivada

Se ha colocado una pieza cúbica para comprobar si el funcionamiento de SC_Pinza es correcto. Como puede verse, tanto “Cierra” como “Cogida” tienen el valor 0. Si ahora pulsamos “Cierra” ocurre esto:

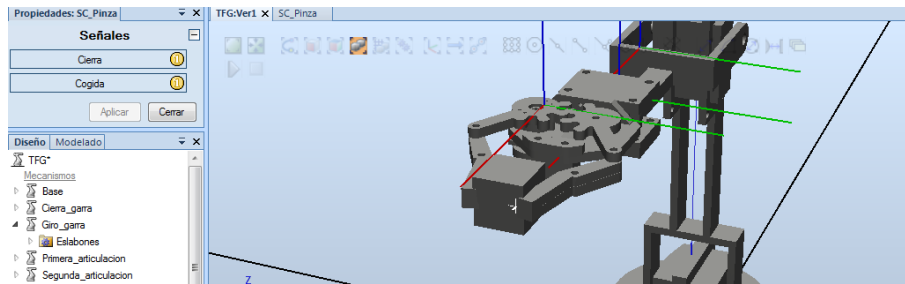


Figura 4-49. Propiedades de SC_Pinza: Cierra activada

Vemos que tanto “Cierra” como “Cogida” tienen el valor 1, ya que la garra se ha cerrado y en su camino ha encontrado una pieza, por lo que la pieza se conecta a la garra y se activa “Cogida”. Si moviésemos el brazo mediante los mecanismos que diseñamos en el apartado anterior, la pieza se movería coherentemente con ellos. Si volvemos a pulsar “Cierra” volveríamos a la situación de la figura 4-46, con “Cierra” y “Cogida” con valor 0 y la pieza desconectada de la garra, por lo que podemos afirmar que el SC_Pinza funciona tal y como deseábamos.

4.3 Controlador del robot

Para dar el siguiente paso, el poder realizar movimientos, programar trayectorias, etc. debemos crear un controlador asociado al robot. Para ello, dentro de la pestaña “Posición inicial”, pulsamos “Sistema del robot” y tenemos tres opciones: “Desde diseño”, “Nuevo sistema” y “Sistema existente”

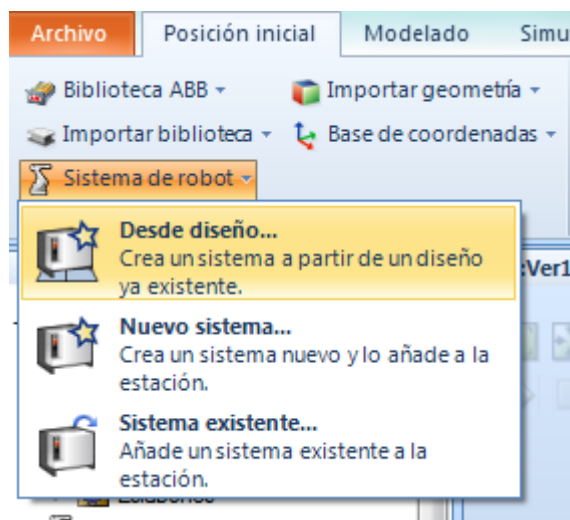


Figura 4-50. Creación de controlador

Pues bien, lo cierto es que no es posible crear un controlador para nuestro brazo, independientemente de la opción que escojamos. Esto es debido a que, a la hora de crear un controlador, en cada una de las opciones se toma de base uno de los modelos de robot que vienen incluidos en la biblioteca ABB y, al no corresponderse estos con nuestro brazo, siempre falla al intentar crear el controlador. Hemos de suponer que, al ser un programa de ABB, está preparado para funcionar con sus propios modelos de robots, así que no podremos tener acceso a las funciones que pretendíamos, como por ejemplo la definición de trayectorias.

5 CONCLUSIONES

*La voluntad obstinada de perseguir una ambición propia es
verdaderamente una fuerza que puede hacer superar
obstáculos.*

- Enzo Ferrari -

Para este trabajo se ha tenido que diseñar un brazo robótico el cual posteriormente sería impreso mediante una impresora 3D. Una de las cosas que más dificultad me ha supuesto ha sido el diseño de cada pieza, ya que si una pieza no tenía sus dimensiones correctamente asignadas puede suceder que esa pieza no pueda cumplir su función correctamente (como de hecho ha sucedido con alguna pieza, que por una cuestión de milímetros ha dado problemas). Por otra parte, la impresión 3D es una tecnología que brinda muchas posibilidades de cara al futuro, por lo que ha sido muy enriquecedor familiarizarse y poder hacer uso de ella.

En cuanto a Arduino, gracias a sus características es gratificante y relativamente sencillo poder crear tus propios proyectos de electrónica y control, pese a estar un poco limitado. Me ha parecido interesante el hecho de haber trabajado con joysticks, ya que son elementos cotidianos que usamos a menudo (para los videojuegos, por ejemplo) pero que, gracias a este trabajo he podido ver como funcionan en detalle.

Por último, he tenido que aprender a usar Robotstudio (o, al menos, un poco, ya que el programa ofrece muchas opciones, y varias de ellas no hemos podido usarlas). Creo que es un programa potente, con el que se pueden hacer simulaciones de una estación industrial con bastante detalle, con lo que me alegro de haberlo usado, aunque, lamentablemente, debido a lo que nos ocurrió al final del trabajo, no pudimos ahondar más en sus múltiples opciones y herramientas

REFERENCIAS

·Material del curso de Robotstudio de la asignatura Control de robots manipuladores, del Máster en Ingeniería automática, robótica y telemática de la Universidad de Sevilla, impartido por el profesor David Muñoz.

·Curso de Robotstudio de 7 temas de José Angel Alonso. Enlace:

<https://www.youtube.com/playlist?list=PLgwW4GLuvZFGjfaZ90RTqxjYaw3DWJ9Ji>

·Página web de diseños para impresión 3D:

<https://www.thingiverse.com/>

· Diseño, programación y simulación de estaciones robotizadas industriales con Robotstudio (Agustín Ramos Hurtado)

·ROBOTICA Manipuladores y robots móviles (Aníbal Ollero Baturone)

